Final Report

OPTIMAL GUIDANCE AND CONTROL IN A MISSILE DEFENSE SYSTEM

By. R. E. LARSON R. M DRESSLER W. G. KECKLER D. G. LUENBERGER
L MEIER R S RATNER A. SHEFI

Prepared for.

NIKE-X PROJECT OFFICE
U.S ARMY MATERIEL COMMAND
REDSTONE ARSENAL, ALABAMA 35809

CONTRACT DA-01-021-AMC-90006(Y)

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

*SRI

Best
Available
Copy

STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA

December 1967

Final Report

OPTIMAL GUIDANCE AND CONTROL IN A MISSILE DEFENSE SYSTEM

By: R. E. LARSON R. M. DRESSLER W. G. KECKLER D. G. LUENBERGER
L. MEIER R. S. RATNER A. SHEFI

Prepared for:

NIKE-X PROJECT OFFICE
U.S. ARMY MATERIEL COMMAND
REDSTONE ARSENAL, ALABAMA 35809

CONTRACT DA-01-021-AMC-90006(Y)

SRI Project 5188-233

Approved: P. E. MERRITT, MANAGER
INFORMATION AND CONTROL LABORATORY

J. D. NOE, EXECUTIVE DIRECTOR
INFORMATION SCIENCE AND ENGINEERING

Copy No. 102

ABSTRACT

This report summarizes the work done by the Information and Control Laboratory during 1967 in the area of optimal guidance and control in a missile defense system. General problem formulations for the optimum interception of both ballistic and maneuvering reentry vehicles are given. A computer program based on dynamic programming for prelaunch calculations is described. A second computer program that utilizes the gradient method for in flight guidance is also discussed. Finally, a game theoretic approach to the problem of intercepting maneuvering reentry vehicles is presented.

CONTENTS

ABSTRACT iii

LIST OF ILLUSTRATIONS. vii

LIST OF TABLES viii

Chapter I FUNDAMENTAL PROBLEM FORMULATIONS. 1
- A. Introduction. 1
- B. Formulation of the Ballistic Missile Interception Problem. 2
  - 1. General Formulation 2
  - 2. General Solution. 7
  - 3. Solution Prior to Launch. 10
  - 4. Solution After Launch 18
- C. Interception of an MRV. 21
- D. Contents of the Remainder of this Report. 24
- References. 27

Chapter II APPLICATION OF DYNAMIC PROGRAMMING TO THE PRELAUNCH CALCULATION 29
- A. Introduction. 29
- B. Dynamic Programming 31
  - 1. Problem Formulation for the Deterministic Case. . . . 31
  - 2. Derivation of the Basic Iterative Functional Equation 32
  - 3. The Standard Computational Algorithm. 33
- C. Forward Dynamic Programming 34
- D. Formulation of the Minimum-Time Problem 40
- E. The Forward Dynamic Programming Solution. 44
- F. Results 45
- G. Conclusions 47
- References. 49

Chapter III APPLICATION OF GRADIENT METHODS TO THE AMM GUIDANCE PROBLEM. 51
- A. Introduction. 51
- B. AMM Equations of Motion 52
- C. Problem Formulation 54
- D. Gradient Method 57
- E. Results 60
- F. Conclusions and Summary 76
- References 77

CONTENTS

Chapter IV		APPLICATION OF A GAMING APPROACH TO MRV INTERCEPTION	79
	A.	Introduction	79
	B.	Problem Formulation.	80
	C.	Existence of Solutions	83
	D.	A Simplified Case.	84
	E.	Examples	85
	F.	Conclusions.	91

ILLUSTRATIONS

Chapter I	Fig. 1	Flow Chart for Solution Prior to Launch	17
	Fig. 2	Flow Chart for Solution After Launch	21
Chapter II	Fig. 1	First Step in Forward Dynamic Programming Procedure	36
	Fig. 2	Tentative Minimum Costs at $k = 2$	37
	Fig. 3	Complete Results at $k = 2$	37
	Fig. 4	Complete Results for Forward Dynamic Programming Example	38
	Fig. 5	Optimal Trajectory if Final State Must Be $x = 2$	38
	Fig. 6	Optimal Trajectory if Final State Is Not Constrained	39
	Fig. 7	Missile Geometry	41
	Fig. 8	Representative Minimum-Time Trajectory to $(y, z) = (131000, 12900)$	46
Chapter III	Fig. 1	AMM Geometry	52
	Fig. 2(a)	Angle-of-Attack Histories for Case 1	62
	Fig. 2(b)	Trajectories in the y-z Plane for Case 1	63
	Fig. 2(c)	Cost *vs.* Iterations for Case 1	64
	Fig. 3(a)	Angle-of-Attack Histories for Case 2	65
	Fig. 3(b)	Trajectories in the y-z Plane for Case 2	66
	Fig. 3(c)	Cost *vs.* Iterations for Case 2	67
	Fig. 4(a)	Angle-of-Attack Histories for Case 3	68
	Fig. 4(b)	Trajectories in the y-z Plane for Case 3	69
	Fig. 4(c)	Cost *vs.* Iterations for Case 3	70
	Fig. 5(a)	Angle-of-Attack Histories for Case 4	71
	Fig. 5(b)	Trajectories in the y-z Plane for Case 4	72
	Fig. 5(c)	Cost *vs.* Iterations for Case 4	73
	Fig. 6	Results for Case 5: Miss Distance *vs.* Crossing Angle	74
Chapter IV	Fig. 1	Available Controls for AMM and Attacker	86
	Fig. 2	Set of Attacker Trajectories Where Initial State is Four Units Above Target	86
	Fig. 3	Example 1: One AMM Where Interception Must Occur at Least Two Units Above Ground	87
	Fig. 4	Three Basic Mixed Strategies for Attacker (Example 1)	88
	Fig. 5	Example 2: One AMM Where Interception Must Occur at Least One Unit Above Ground	88

ILLUSTRATIONS

Chapter IV	Fig. 6	Interceptor Optimal Control Law When Interceptor Remains on Ground for Two Units of Time	89
	Fig. 7	Six Basic Mixed Strategies for Attacker (Example 2)	89
	Fig. 8	Example 3: Two AMMs Located on Target Where Interception Must Occur at Least One Unit Above Ground.	90
	Fig. 9	Interceptor Optimal Control Law When Both AMMs Remain on Ground for Two Units of Time (Example 3)	91
	Fig. 10	Three Basic Mixed Strategies for Attacker (Example 3)	91

TABLES

Chapter II	Table I	Values of Terminal Cost Function and the Total Cost of Terminating in Each Quantized State	40
Chapter III	Table I	Target Perameters for the Computer Runs	61

Chapter I

FUNDAMENTAL PROBLEM FORMULATIONS

A. Introduction

The purpose of this report is to present the results of work done during 1967 by the Information and Control Laboratory in support of NIKE-X System Evaluation Studies. The present report covers the area of optimal guidance and control, while a second report discusses optimum estimation[1] *

The basic approach taken in these studies is to treat the intercept problem using the methods of optimal control theory. In a previous report on this contract[2], the intercept problem was formulated in this framework. There it was shown that, if all random effects are explicitly taken into account, the interception of a ballistic reentry vehicle becomes a special case of the combined optimal control and estimation problem[3]. It can also be shown that the problem of intercepting a maneuvering reentry vehicle (MRV) is a special case of a stochastic differential game[4]. It is recognized that the solution of the general case of either of these problems is not computationally feasible. However, in Ref. 2 it was pointed out that, by taking advantage of the particular structure of the intercept problem, it might be possible to reduce computational requirements to the point where a realistic solution could be obtained on a computer facility with the capability of the proposed NIKE-X computer system. This report summarizes the progress that has been made toward this end.

In the next section of this chapter the problem of intercepting a ballistic target is considered. First, a general problem formulation is given. Then, a solution for this general case is presented. Next, the nature of the problem before the AMM has been launched is considered in greater detail; a computationally feasible solution based on assumptions that are generally met in practice is presented. Finally, the

* References are listed at the end of each chapter.

situation after the AMM has been launched is examined; a realistic yet computationally feasible solution for this case is given.

In the following section the interception of an MRV is briefly considered. Emphasis is placed on the differences from the ballistic case and how these can be explicitly taken into account. Hard-point defense and use of multiple interceptors are both discussed.

In the final section of this chapter the remainder of the report is outlined. Each of the three remaining chapters discusses computer programs that have been written for solving the three basic problems indicated above. In Chap. II a program based on dynamic programming for the case of intercepting a ballistic target when the AMM has *not* been launched is discussed. Chapter III presents a program based on the gradient method for the case of intercepting a ballistic target when the AMM has already been launched. In the final chapter, a game-theoretic approach to the MRV problem is described.

B. Formulation of the Ballistic Missile Interception Problem

1. General Formulation

The problem of intercepting a ballistic target can be formulated in the following manner. First, the system equations for the ballistic target are written in state-space notation as

$$\dot{\underline{x}}_T = \underline{f}_T(\underline{x}_T, t) + \underline{w}_T \qquad (1)^*$$

where

$\underline{x}_T$ = State vector for target

$\underline{w}_T$ = Random forcing function vector for target with known probability density function, $p(\underline{w}_T)$

t = Time

$\underline{f}_T$ = Vector function representing the external deterministic forces on target

$(\dot{\ }) = d/dt\ ()$

---

* All equations are renumbered for each section, as are figures and tables.

In Ref. 5 the seven dimensional state vector contains three Cartesian position coordinates (x, y, and z), three Cartesian velocity coordinates ($\dot{x}$, $\dot{y}$, and $\dot{z}$), and the ratio of air density to ballistic coefficient (ρ/β). As discussed in Ref. 5, the ballistic coefficient is included in a state variable because it cannot be determined *a priori* for an unknown target. The function f_T is then obtained by defining drag and gravity to be the only external forces on the target and by taking air density to be a locally exponential function of altitude. Explicit formulas appear in Sec. III of Ref. 5. The statistics of the random forcing function vector are determined by approximating the magnitude of external forces not accounted for in f_T; for this case the probability density function for u_T is gaussian with mean zero and covariance matrix as in Sec. IV of Ref. 5.

The state vector is not directly known. Instead, a vector of measurements to which noise has been added is available. The measurements are a function of some, but not necessarily all, of the state variables. The measurement equation is

$$\underline{z}_T = \underline{h}_T(\underline{x}_T) + \underline{\nu}_T \tag{2}$$

where

$\underline{z}_T$ = Measurement vector for target

$\underline{\nu}_T$ = Measurement noise vector for target with known probability density function, $p(\underline{\nu}_T)$

$\underline{h}_T$ = Vector function representing the measurements in terms of the state variables of the target

In Ref. 5 the three-dimensional measurement vector is taken to be three position measurements in radar coordinates (range and two radar angles). The function $\underline{h}_T$ is determined from geometrical considerations, as in Sec. III of Ref. 5. The measurement noise vector for this case is based on a model of the MSR described in Ref. 6, the probability density function for $\underline{\nu}_T$ is gaussian with a mean determined by bias terms and with the covariance matrix shown in this reference.

The time at which computations begin is denoted as t_0. At this time an *a priori* probability density function of the target state vector, $p_0[\underline{x}_T(t_0)]$, is known. In Ref. 5 this probability density function is determined by the track initialization procedure; it is gaussian with mean and covariance matrix determined as in Sec. IV of Ref. 5.

The AMM is treated in a similar manner. The system equations are

$$\dot{\underline{x}}_A = \underline{f}_A(\underline{x}_A, \underline{u}_A, t) + \underline{w}_A \tag{3}$$

where

$\underline{x}_A$ = State vector for AMM

$\underline{u}_A$ = Deterministic control vector applied to AMM

$\underline{w}_A$ = Random forcing function vector for AMM with known probability density function $p(\underline{w}_A)$

$\underline{f}_A$ = Vector function representing the external forces on the AMM.

In this report the six-dimensional state vector $\underline{x}_A$ contains three Cartesian position coordinates (x, y, and z) and three Cartesian velocity coordinates ($\dot{x}, \dot{y}$, and $\dot{z}$). Because the aerodynamic characteristics of the AMM can be studied in advance, ballistic coefficient need not be a state variable, but instead can be a tabulated function of the other state variables.

The deterministic control vector contains all parameters that affect the AMM trajectory and that can be commanded directly. For an AMM with aerodynamic steering only, this vector contains two components which specify the angle of attack in three-dimensional space. If the thrust is controlled, it also is added to the control vector; otherwise, it is a tabulated function of time. Throughout this report, it is assumed that the thrust is not controlled; however, all of the techniques discussed here can easily be extended to the more general case where the thrust is controlled. It is also assumed that no guidance commands are given during the operation of first stage. The control vector for the entire duration of first stages is taken to be two parameters that determine the angle between the local vertical and the vector from the missile site to AMM position at the end of first stage. For any chosen angle the AMM follows a pre-programmed angle-of-attack sequence to reach this point.

The function $\underline{f}_A$ is determined by writing expressions for the external forces on the AMM, namely drag, lift, thrust, and gravity, and expressing them in terms of system state variables and/or tabulated functions of these variables and time.

It is possible to specify a probability density function for the random forcing function vector, $\underline{w}_A$. However, in most applications it is sufficient to set $\underline{w}_A$ to zero.

If desired, it is possible to stipulate that only noisy measurements of the AMM state vector are available. In this case the measurement equation is

$$\underline{z}_A = \underline{h}_A(\underline{x}_A, t) + \underline{v}_A \tag{4}$$

where

$\underline{z}_A$ = Measurement vector for AMM

$\underline{v}_A$ = Measurement noise vector for AMM with known probability density function, $p(\underline{v}_A)$

$\underline{h}_A$ = Vector function representing the measurements in terms of the state variables of the AMM.

However, in many practical cases it is possible to assume that the state vector of the AMM is directly known, *i.e.*, that $\underline{z}_A = \underline{x}_A$.

At t_0 the probability density function of the state vector of the AMM, $p_0[\underline{x}_A(t_0)]$, is specified. Again, it is generally assumed that $\underline{x}_A(t_0)$ is known. If the missile has not yet been launched, then $\underline{x}_A(t_0)$ consists of position at the launch site and velocity zero

The general performance criterion includes two types of terms. The first term is an integral reflecting the cost incurred during the flight of the AMM. This integral criterion is written

$$J_1 = E\left\{\int_{t_l}^{t_f} l[\underline{x}_A(\sigma), \underline{u}_A(\sigma), \sigma]\, d\sigma\right\} \tag{5}$$

where

J_1 = Cost on flight of AMM

l = Cost per unit time on flight of AMM

t_l = AMM launch time

t_f = Intercept time; firing time of AMM warhead

σ = Dummy variable for time

E = Expected value operator.

The expectation must be taken if a random forcing function and/or measurement noise affects the state of the AMM. One choice of this criterion is a minimum-time-of-flight, in which case $l = 1$ and $J_1 = t_f - t_l$.

The second term is a terminal cost based on the state vectors of the target and the AMM at intercept time. This term can be expressed as

$$J_2 = E\{\psi[\underline{x}_A(t_f), \underline{x}_T(t_f), t_f]\} \tag{6}$$

where

J_2 = Expected terminal cost of an engagement

ψ = Terminal cost function of an engagement.

The expectation must be taken because of the random forcing function and measurement noise that affect the target and possibly also the AMM. The terminal cost function itself generally consists of two types of terms. The first type indicates how successful the intercept is; typical choices include kill probability, miss distance or crossing angle. The second type reflects where the intercept occurs; this could be altitude of intercept or some other function of battle space conserved.

Constraints are imposed on the ballistic target and the AMM by physical limitations, such as maximum acceleration, maximum aerodynamic heating, and maximum angle of attack, as well as by operational considerations such as fratricide avoidance. These constraints can be written

$$\begin{aligned} \underline{x}_T &\in X_T \\ \underline{x}_A &\in X_A \\ \underline{u}_A &\in U_A \end{aligned} \tag{7}$$

where

X_T = Set of admissible values of the ballistic target state vector

X_A = Set of admissible values of the AMM state vector

U_A = Set of admissible values of the AMM deterministic control vector

The sets X_T and X_A can vary with time, while the set U_A can vary with both $\underline{x}_A$ and time.

The problem can now be formulated in the following way:

Given:

(1) Dynamic equations for the ballistic target and AMM

$$\dot{\underline{x}}_T = \underline{f}_T(\underline{x}_T, t) + \underline{w}_T$$

$$\dot{\underline{x}}_A = \underline{f}_A(\underline{x}_A, \underline{u}_A, t) + \underline{w}_A$$

(2) Measurement equations for the ballistic target and AMM

$$\underline{z}_T = \underline{h}_T(\underline{x}_T, t) + \underline{v}_T$$

$$\underline{z}_A = \underline{h}_A(\underline{x}_A, t) + \underline{v}_A$$

(3) Statistics of the random forcing functions, the measurement noises, and the *a priori* states of the ballistic target and the AMM

$$p(\underline{v}_T), \qquad p(\underline{w}_T), \qquad p_0[\underline{x}_T(t_0)]$$

$$p(\underline{v}_A), \qquad p(\underline{w}_A), \qquad p_0[\underline{x}_A(t_0)]$$

(4) All measurements of the ballistic target and AMM up to the present time, t_p

$$\underline{z}_T(t), \qquad t_0 \leq t \leq t_p$$

$$\underline{z}_A(t), \qquad t_0 \leq t \leq t_p$$

(5) A performance criterion

$$J = J_1 + J_2 = E\{\int_{t_l}^{t_f} l[\underline{x}_A(\sigma), \underline{u}_A(\sigma), \sigma]d\sigma + \psi[\underline{x}_A(t_f), \underline{x}_T(t_f), t_f]\}$$

(6) Constraints on the ballistic target and the AMM

$$\underline{x}_T \in X_T$$

$$\underline{x}_A \in X_A$$

$$\underline{u}_A \in U_A$$

Find

A launch time, t_l, a firing time, t_f, and a control policy, $\underline{u}_A(\sigma)$, $t_l \leq \sigma \leq t_f$, such that the performance criterion in (5) is minimized, where the expectation is conditioned on all past measurements and *a priori* statistics, and where the constraints in (6), the dynamic equations in (1), and the measurement equations in (2) are all satisfied.

2 General Solution

The problem formulated in the previous section is a combined optimal control and estimation problem[3]. A general solution to this problem can be obtained by using dynamic programming[3,7,8]. For simplicity it will be

assumed in this section that the launch time, t_l, and the intercept time, t_f, are fixed.* The first step in this procedure is to assume that all observations and controls occur at discrete times, t_0, $t_0 + \Delta t$, $t_0 + 2\Delta t$, ..., $t_0 + K\Delta t = t_f$. For simplicity in notation, the time instants $t_0 + k\Delta t$ are then abbreviated as k. Also, the sequence of observations up to time k is denoted as

$$Z_k = \{\underline{z}_T(k), \underline{z}_A(k), \underline{z}_T(k-1), \underline{z}_A(k-1), \ldots, \underline{z}_T(0), \underline{z}_A(0)\} \quad . \tag{8}$$

The sequence of past controls is written as

$$U_k = \{\underline{u}_A(k), \underline{u}_A(k-1), \ldots, \underline{u}_A(0)\} \quad . \tag{9}$$

The differential equations, Eqs. (1) and (3), are then replaced by the following difference equations:

$$\underline{x}_T(k+1) = \underline{x}_T(k) + \underline{f}_T[\underline{x}_T(k), k]\Delta t + \underline{w}_T(k)\Delta t$$

$$\underline{x}_A(k+1) = \underline{x}_A(k) + \underline{f}_A[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t + \underline{w}_A(k)\Delta t \quad . \tag{10}$$

The integral in Eq. (5) is approximated by the summation

$$J_1 = \sum_{k=k_l}^{K} l[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t \tag{11}$$

where k_l corresponds to launch time, *i.e.*, $k_l\Delta t = t_l - t_0$.

Now, the minimum cost function is defined for all k, $k = 0, 1, \ldots K$, and for all possible sequences of *past* measurements and controls as

$$I(Z_k; U_{k-1}; k) = \min_{\underline{u}_A(k), \ldots, \underline{u}_A(K)} \left(E\left\{ \sum_{k=0}^{K} l[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t + \psi[\underline{x}_T(K), \underline{x}_A(K), K] \right\} \right) \tag{12}$$

where the expectation is over $\underline{x}_T(k)$, $\underline{x}_A(k)$, $\underline{w}_T(j)$, and $\underline{w}_A(j)$, $j = k$, $k+1, \ldots, K$; $\underline{v}_T(j)$ and $\underline{v}_A(j)$, $j = k+1, \ldots K$ *conditioned* on Z_k

---

*The determination of t_l and t_f requires a further search over the solutions obtained here. The nature of this search will be discussed in connection with specific problems in the next two sections.

and U_{k-1}. Using the approach of Ref. 3 and assuming that the statistical variables $[w_A, w_T, v_A, v_T, x_A(0)$, and $x_T(0)]$ are uncorrelated with each other and with themselves at different time instants,* the basic iterative equation can be derived as

$$I(Z_k;U_{k-1};k) = \underset{u_A(k)}{\mathrm{Min}} \Big\{ E \Big[l[x_A(k), \underline{u}_A(k), k]\Delta t$$

$$+ I\Big(\underline{h}_T\{\underline{x}_T(k) + \underline{f}_T[\underline{x}_T(k), k]\Delta t + \underline{w}_T(k)\Delta t, k\} + \underline{v}_T(k+1),$$

$$\underline{h}_A\{\underline{x}_A(k) + \underline{f}_A[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t + \underline{w}_A(k)\Delta t, k\} + \underline{v}_A(k+1),$$

$$Z_k, \underline{u}_A(k), U_{k-1}; k \Big) \Big] \Big\} \qquad (13)$$

where the expectation is now over $\underline{x}_T(k)$, $\underline{x}_A(k)$, $\underline{w}_T(k)$, $\underline{w}_A(k)$, $\underline{v}_T(k+1)$, and $v_A(k+1)$, again conditioned on Z_k and U_{k-1}. In the expectation, $w_T(k)$, $\underline{w}_A(k)$, $\underline{v}_T(k+1)$, and $\underline{v}_A(k+1)$ are independent of Z_k and U_{k-1}, so that the unconditional probability density functions can be used. However, the probability density functions for $\underline{x}_T(k)$ and $\underline{x}_A(k)$ depend very strongly on these quantities. The conditional probability density functions can be computed by iterative application of Bayes' rule. For $\underline{x}_T(k)$ the relation is

$$p[\underline{x}_T(j+1)/Z_{j+1}, U_j]$$

$$= \frac{p[\underline{z}_T(j+1)/\underline{x}_T(j+1)] \int_{\underline{x}_T(j)} \Big\{ p[\underline{x}_T(j+1)/\underline{x}_T(j)]\, p[\underline{x}_T(j)/Z_j, U_{j-1}] \Big\} d\underline{x}_T(j)}{\int_{\underline{x}_T(j+1)} p[\underline{z}_T(j+1)/\underline{x}_T(j+1)] \int_{\underline{x}_T(j)} \Big\{ p[\underline{x}_T(j+1)/x_T(j)]\, p[\underline{x}_T(j)/Z_j, U_{j-1}] \Big\} d\underline{x}_T(j)\, d\underline{x}_T(j+1)}$$

$$j = 0, 1, \ldots k-1 \quad , \qquad (14)$$

where

$$\int_{\underline{x}_T(j)} \ldots \, d\underline{x}_T(j)$$

denotes integration over all possible values of $\underline{x}_T(j)$. The probability density function $p[\underline{z}_T(j+1)/\underline{x}_T(j+1)]$ is found from Eq. (2) and

---

* These assumptions can be relaxed at the expense of defining additional state variables to account for correlation.

knowledge of $p[\underline{v}_T(j+1)]$. The probability density function $p[\underline{x}_T(j+1)/\underline{x}_T(j)]$ is determined from Eq. (1) and knowledge of $p[\underline{w}_T(j)]$. The probability density function $p[\underline{x}_T(j)/Z_j, U_{j-1}]$ is known from the previous iteration. In order to begin the iterations, the *a priori* probability density function, $p_0[\underline{x}_T(0)]$, is used. Formally,

$$p[\underline{x}_T(0)/Z_0, U_{-1}] = p_0[\underline{x}_T(0)] \quad . \tag{15}$$

An equation similar to Eq. (14) can be written to determine the conditional probability density function for $\underline{x}_A(k)$. The only difference is that $p[\underline{x}_T(j+1)/\underline{x}_T(j)]$ is replaced by $p[\underline{x}_A(j+1)/\underline{x}_A(j), \underline{u}_A(j)]$, which is then determined from Eq (3) and knowledge of both $\underline{u}_A(j)$ and $p[\underline{w}_A(j)]$.

In order to start the iterations in Eq. (13),

$$\iota(Z_K; U_{K-1}; K) = E\{\psi[\underline{x}_T(K), \underline{x}_A(K), K]\} \tag{16}$$

where the expectation is over $x_T(K)$ and $x_A(K)$ conditioned on Z_K and U_{K-1}.

This solution is extremely difficult to implement in the general case, primarily because a new quantity is added to the argument of the minimum cost function every time a measurement is received or a control is applied. The dimensionality of the problem thus increases with time; and even a problem with relatively low-dimensional state, control, and measurement vectors quickly becomes unmanageable.

3. Solution Prior to Launch

If the AMM has not yet been launched, then the launch time is determined as part of this procedure. However, in an actual engagement, the launch decision must be based to some extent on the present positions of the other interceptors and targets. In this section it will be assumed that a supervisory program, referred to here as the launch doctrine, will examine the solution and determine whether or not an immediate launch should be made. Methods for enabling the launch doctrine to make the most effective use of the information obtained by this solution are currently under study and will be discussed in a future memorandum.

Because of the specific nature of the intercept problem, it is possible to utilize the results of Ref. 9 to reduce considerably the amount of computational effort required for a complete solution. In

this reference it is shown that the procedure of the previous section can be broken down into a number of simpler functions with no loss of generality.

The first step in the solution method of the previous section is to obtain the conditional probability density functions of the target state and the AMM state at the present time. In the case of the AMM, because it is still at the launch site, no estimation is required. The AMM state is known to be, with probability one,

$$\underline{x}_A(k) = \begin{bmatrix} x_{1s} \\ x_{2s} \\ x_{3s} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

where x_{1s}, x_{2s}, x_{3s} are the Cartesian coordinates of the launch site and where the zeros indicate that the AMM has zero initial velocity.

With the state of the AMM completely determined, estimation of the target state can be performed independently. Previous work[5] has shown that the extended Kalman filter is quite suitable for this function. The use of this filter assumes that the conditional probability density function for the target state can be represented as a multidimensional gaussian distribution; formally,

$$p[\underline{x}_T(k) \mid Z_k] = N[\hat{\underline{x}}_T(k/k), S_T(k/k)] \tag{18}$$

where $\hat{\underline{x}}_T(k/k)$, the conditional mean, and $S_T(k/k)$, the conditional covariance matrix, are determined recursively as in Ref. 5. A new calculation of these quantities is performed each time a new measurement, $\underline{z}_T(k)$, is received. For this reason the filter is particularly well-suited for on-line estimation.

Under the assumptions of Ref. 5, the recursive equations can be written as a set of prediction equations,

$$\hat{\underline{x}}_T(k/k-1) = \hat{\underline{x}}_T(k-1/k-1) + \underline{f}_T[\hat{\underline{x}}_T(k-1/k-1), k]\Delta t \quad ,$$

$$S_T(k/k-1) = \Phi_T(k-1)S_T(k-1/k-1)\Phi_T^T(k-1) + Q_T(k-1) \tag{19}$$

where

$$\Phi_T(k) = I + F_T(k)\Delta t$$

$$F_{T_{ij}}(k) = \left.\frac{\partial f_{T_i}}{\partial x_{T_j}}\right|_{\hat{\underline{x}}_T(k/k)}$$

$$Q_T(k) = \text{Cov}\ [\underline{w}_T(k)] \quad ,$$

and a set of correction equations

$$\hat{\underline{x}}_T(k/k) = \hat{\underline{x}}_T(k/k-1) + W_T(k)\{\underline{z}_T(k) - \underline{h}_T[\hat{\underline{x}}_T(k/k-1),\ k]\}$$

$$S_T(k/k) = [I - W_T(k)H_T(k)]S_T(k/k-1) \tag{20}$$

where

$$W_T(k) = S_T(k/k-1)H_T^T(k)[R_T(k) + H_T(k)S_T(k/k-1)H_T^T(k)]^{-1}$$

$$H_T(k) = \left.\frac{\partial h_{T_i}}{\partial x_{T_j}}\right|_{\hat{\underline{x}}_T(k/k-1)}$$

$$R_T(k) = \text{Cov}\ [\underline{v}_T(k)] \quad .$$

In order to start the recursive equations, the *a priori* probability density function of $\underline{x}_T$ is taken to be gaussian with mean $\hat{\underline{x}}(0/0)$ and covariance matrix $S_T(0/0)$. Formally,

$$p_0[x_T(0)] = N[\hat{\underline{x}}(0/0), S_T(0/0)] \quad . \tag{21}$$

The computation of optimal stochastic control can also be separated into several simpler calculations. First, it is seen that the control $\underline{u}_A(k)$ has no effect whatever on the target state. Thus, the probability density function of the target state at future times is completely specified by knowing the probability density function at the present time and the target dynamic equations. If the assumptions of Ref. 5 hold, then the probability density function of future target states is again gaussian[10], formally,

$$p[\hat{x}_T(j), Z_k] \quad N[\hat{x}_T(j/k), S_T(j/k)] \tag{22}$$

where

$$j \le k \quad .$$

The mean and covariance matrix again obey recursive relations[10]; for the assumptions of Ref. 5 these equations are

$$\begin{aligned} \hat{\underline{x}}_T(j + 1/k) &= \hat{\underline{x}}_T(j/k) + \underline{f}_T[\hat{\underline{x}}_T(j/k), k]\Delta t \\ S_T(j + 1/k) &= \Phi_T(j) S_T(j/k)\, \Phi_T^T(j) + Q(j) \end{aligned} \tag{23}$$

where

$$j > k \quad .$$

Work is currently being performed to find closed-form expressions for $\hat{x}(j + 1/k)$ and $S_T(j + 1/k)$ for certain forms of $\underline{f}_T$; analytic prediction of these quantities is then possible. Results of this study will be reported in a future memorandum.

At this point it is convenient to assume that there is no uncertainty associated with the state of the AMM. Specifically, it is assumed that $\underline{w}_A \equiv 0$, $\underline{z}_A = \underline{x}_A$, and $\underline{v}_A \equiv 0$. Because the initial state of the AMM is known, because the AMM dynamics can be studied in advance, and because a radar beacon can be mounted on the AMM, it is clear that the uncertainty of the state of the AMM will be considerably less than that associated with the target state. Thus, the approximation that the AMM state is known exactly is justifiable from an engineering point of view.

It is now appropriate to reexamine the problem to be solved. The quantity that is to be minimized is still the sum of J_1 and J_2, as expressed in Eq (2). However, the expectation is now to be taken only over $\underline{x}_T(K)$. Furthermore, since launch time and intercept time are to be chosen, the minimization is not only over $\underline{u}_A(k)$, $k = k_l, \ldots, K$, but also over k_l and K themselves. The problem thus becomes that of finding J', where

$$J' = \underset{k_l; K; \underline{u}_A(k), k=k_l, \ldots K}{\mathrm{Min}} \quad \underset{x_T(K)}{E} \left\{ \sum_{k=k_l}^{K} l[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t + \psi[\underline{x}_T(K), \underline{x}_A(K), K] \right\} . \tag{24}$$

The equations of motion for the AMM are

$$\underline{x}_A(k + 1) = \underline{x}_A(k) + \underline{f}_A[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t \quad , \tag{25}$$

where $\underline{x}_A(k)$ is always known exactly. The constraints on the AMM are, as before

$$\underline{x}_A \in X_A$$
$$\underline{u}_A \in U_A \quad . \tag{26}$$

The probability density function of the target state for *any* intercept time, K is known as

$$p[\underline{x}_T(K)/Z_k] = N[\hat{\underline{x}}_T(K/k), S_T(K/k)] \tag{27}$$

where $\hat{\underline{x}}_T(K/k)$ and $S_T(K/k)$ are determined from Eq. (23).

In solving for J', it is useful to consider the following subproblem:

> For any given state at intercept, $\underline{x}_A(K)$, find the control sequence, $\underline{u}_A(k)$, $k = k_l, \ldots K$, such that the cost on the AMM,
>
> $$J_1 = \sum_{k=k_l}^{K} l\,[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t$$
>
> is minimized.

Because the initial AMM state is known to consist of its position at the launch site and velocity zero, and because the optimal trajectory is desired for all possible final states, it is appropriate to consider forward dynamic programming[11] for performing the optimization. The main features of this procedure are summarized in Chap. II and discussed in more detail in Ref. 11. This procedure obtains $I'[\underline{x}_A(K), K]$, the minimum cost of reaching any specified terminal state, $\underline{x}_A(K)$, from the launch site, as well as the function $\hat{\underline{u}}_A[\underline{x}_A(K), K]$, which specifies the corresponding optimal control sequence.

One important feature of forward dynamic programming is that, if the single-state cost function, l, and the dynamic equations, $\underline{f}_A$, depend only on the time that has elapsed after launch and not on absolute time, then the minimum cost function and the optimal control sequence depend only on the state and elapsed time, and not on the actual time. For realistic criteria and for the general AMM dynamic model used, this

is the case, thus, the forward dynamic programming solution can be written as $I'(\underline{x}_A)$, the minimum cost of reaching terminal state $\underline{x}_A$, and $\hat{\underline{u}}_A'(\underline{x}_A)$, the function that determines the corresponding optimal control sequence.

For the problem defined above the computational requirements of forward dynamic programming are such that it is not feasible to obtain a complete solution in real time. However, the nature of the solution is such that the function $I'(\underline{x}_A)$ and $\hat{\underline{u}}_A'(\underline{x}_A)$ can be precomputed off-line, and the solutions recovered in real time. Thus, even though the specification of $I'(\underline{x}_A)$ and $\hat{\underline{u}}_A'(\underline{x}_A)$ may be quite time-consuming, the recovery of minimum costs and corresponding optimal controls can easily be done on-line. Computer programs for both obtaining the solution off-line and recovering it on-line are discussed in detail in Chap. II.

Returning to the original problem, the minimization in Eq. (24) can be rewritten as

$$J' = \min_{K,\,\underline{x}_A} \left(I'(\underline{x}_A) + \mathop{E}_{\underline{x}_T(K)} \{\psi[\underline{x}_T(K), \underline{x}_A, K]\} \right) \tag{28}$$

where $I'(\underline{x}_A)$ has been computed off-line by forward dynamic programming. The minimization is now only over $\underline{x}_A$, the terminal state of the AMM, and K, the intercept time. The optimal control sequence corresponding to any given $\underline{x}_A$ can be recovered from the forward dynamic programming results. The launch time, k_l, is determined from K and this control sequence. If the criterion for this optimization is minimum time, then $K - k_l$ is given directly; otherwise, it can be computed during the optimization and stored along with $I'(\underline{x}_A)$ and $\hat{\underline{u}}_A'(\underline{x}_A)$ (see Chap. II and Ref. 11 for details).

The optimization in Eq. (28) can now be performed by a direct search. First, a search is made over a finite set of values of K, the intercept time. The launch doctrine determines which times to consider, generally, a fairly small number of values. For a given value of K, the optimal choice of $\underline{x}_A$ can be determined by another direct search. First, $p[\underline{x}_T(K)/Z_K]$ is obtained. Then, based on this distribution, a finite set of values of $\underline{x}_A$ is selected. The expected value of ψ is then computed for each value of $\underline{x}_A$ using $p[\underline{x}_T(k)/Z_k]$. The value $I'(\underline{x}_A)$ is also recovered; for each $\underline{x}_A$ the launch time, k_l, is determined from the value of $K - k_l$ corresponding to $I'(\underline{x}_A)$ and the actual value of K. If this value of k_l is at or after the present time, *i.e.*, if the launch time is physically possible then the sum of I' and the expected value of ψ is formed,

if not, then this value of $\underline{x}_A$ is not considered. Finally, the sums for admissible values of $\underline{x}_A$ are compared, and the minimum value is chosen. The minimum values for each K are next compared, and the smallest value is chosen as J'. This completes the direct search over K and $\underline{x}_A$.

The value of J' that is finally obtained is then relayed to the launch doctrine. If the corresponding launch time, k_l, is the present time, then a decision is made on the basis of the entire engagement whether or not to make this particular launch. If the launch time is in the future, the launch doctrine notes this and decides what further computations, if any, need to be made for this case. If the launch doctrine determines that an immediate launch is to be made, then the entire control sequence, $\underline{u}_A(k)$, $k = k_l, \ldots K$, is recovered and transferred to the guidance program. This control sequence is then followed by the AMM until the guidance program determines from subsequent measurements that a correction is required. Procedures for performing this correction are discussed in the next section.

A summary of this procedure appears in the flow chart of Fig. 1. First, the conditional probability density function of the target state at the present time is obtained by use of an extended Kalman filter. Next, this conditional probability density function for future times is found either by numerical integration or analytic prediction. Then, a direct search is made over values of intercept time and the terminal state of the AMM in order to find the minimum value of the performance criterion in Eq. (24). First, the minimum cost of flying the AMM from the launch site to the terminal state is recovered from results that have been precomputed using forward dynamic programming. Next, the expected value of the terminal cost function for the AMM terminal state and the intercept time is evaluated, the expectation is performed with respect to the probability density function of the predicted target state. Finally, the minimum value of the performance criterion is presented to the launch doctrine; if a decision is made to launch, the sequence of guidance commands and the AMM trajectory are recovered from the forward dynamic programming results. The computational requirements of this solution method are such that it can be considered for use in a real-time on-line system. The use of an extended Kalman filter in real time, which is considered in Ref. 5, is currently being studied further. Analytic prediction is clearly feasible in real-time, and the computational requirements of more sophisticated prediction schemes are now being studied.

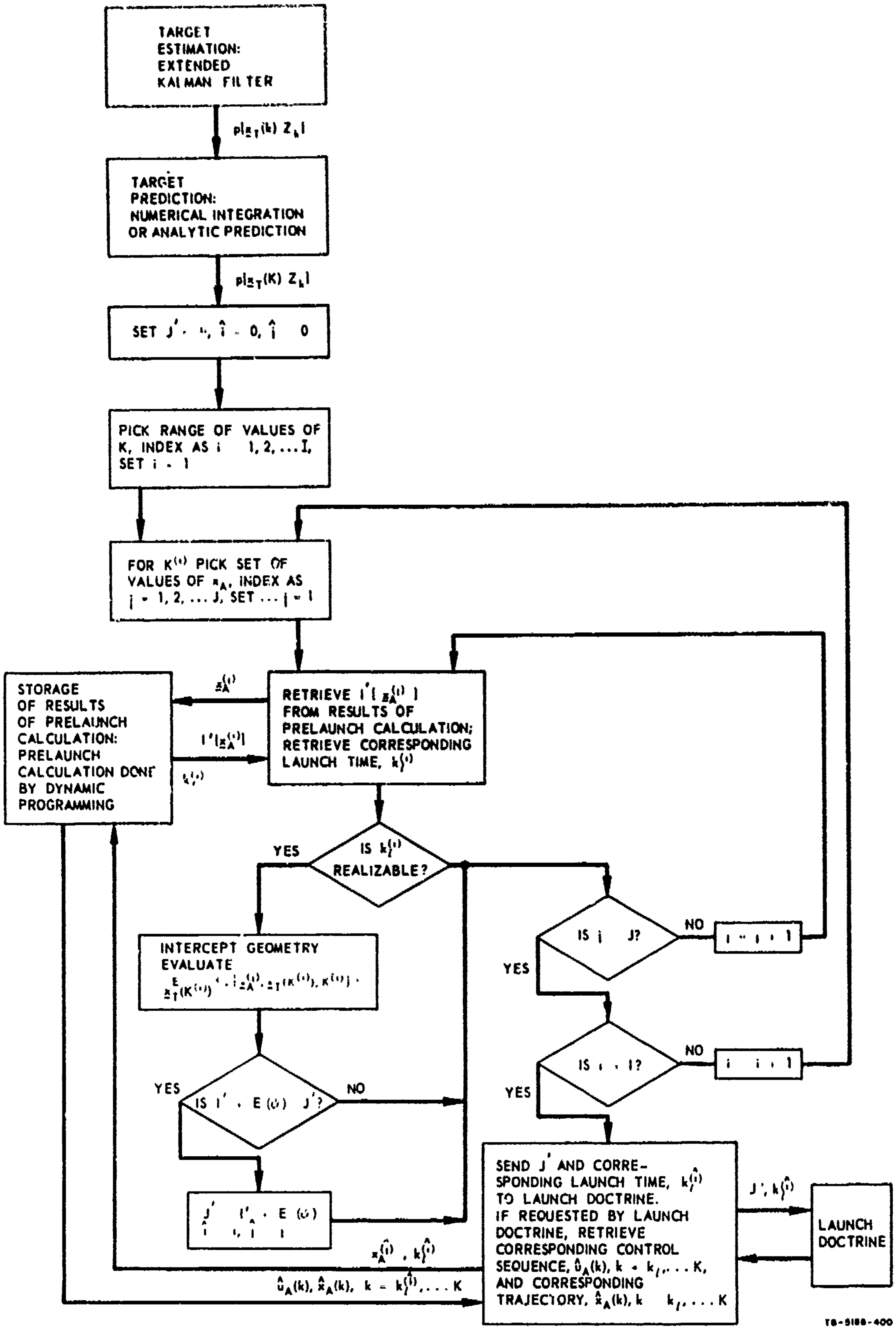

FIG. 1 FLOW CHART FOR SOLUTION PRIOR TO LAUNCH

The complete dynamic programming solution for $I'(x_A)$ requires a considerable amount of time; however, this solution can be precomputed off-line, stored, and recovered in real time. The direct search for minimizing J', including the taking of the expected value of ψ, appears to be feasible for real-time computation, especially if the launch doctrine and the prediction results can be utilized to limit the search to a reasonable number of values.

4. Solution After Launch

Once the AMM has been launched, the nature of the problem will be changed somewhat from that of the previous section. In the first place, the launch time is already determined and cannot be changed. In the second place, the determination of the present position of the AMM is no longer trivial. Nevertheless, the solution can again be broken down into a number of simpler tasks, much as in the previous section.

The probability density function of present and future target states can be handled exactly as in the previous section. Thus, the target estimation and target prediction functions need not be modified at all.

The determination of the present state of the AMM requires at least some computation. It is generally sufficient to assume that this state can be perfectly determined. However, the actual computation of these values requires some data processing. One procedure is to integrate the AMM dynamic equations forward using the guidance commands that have been given. An alternative procedure is to utilize radar measurements of AMM position. Generally, a simple polynomial filter is sufficiently accurate to obtain the complete state, although an extended Kalman filter can be used, if required

When the AMM state has been determined, the optimization problem that remains can be stated, using the notation of previous sections, as follows.

Find:

$$J = \underset{K;\,\underline{u}_A(k),\,k=k_p,\ldots,K}{\text{Min}} \left(\underset{\underline{x}_T(K)}{E} \left\{ \sum_{k=k_p}^{K} l\,[\underline{x}_A(k), \underline{u}_A(k), k] + \psi[\underline{x}_T(K), \underline{x}_A(K), K] \right\} \right) \tag{29}$$

where k_p is the present time, subject to the AMM dynamic equations

$$\underline{x}_A(k+1) = \underline{x}_A(k) + \underline{f}_A[\underline{x}_A(k), \underline{u}_A(k), k]\Delta t \quad , \tag{30}$$

the constraints

$$\underline{x}_A \in X_A$$

$$\underline{u}_A \in U_A \quad , \tag{31}$$

and the initial condition

$$\underline{x}_A(k_p) = \underline{x}_A^{(p)} \tag{32}$$

where $\underline{x}_A^{(p)}$ is the present AMM state.

Because the present AMM state varies throughout the engagement, it is not appropriate to precompute the solution using the forward dynamic programming algorithm discussed in the previous section. Also, because the statistics of the predicted target trajectory are changing during the engagement, a solution precomputed by backward dynamic programming is likewise not feasible. Thus, the optimum intercept problem in Eqs. (29)-(32) must be solved on-line in real time.

An important consideration in solving this problem is the availability of the optimal guidance sequence and the corresponding AMM trajectory for the problem solved at the previous time instant. The only difference between this problem and the present problem is that the present and future statistics of the target state may be modified slightly by the additional measurement received and that the present AMM state is changed. If these differences are small, it is to be expected that the solution to the previous problem is an excellent approximation to that of the present problem.

One method for utilizing the previous solution is a forward dynamic programming procedure in a small region about this optimal trajectory. Because this region covers only a small number of increments in each state variable, the computational requirements can be reduced to the point where an on-line solution is feasible. This approach is discussed in Chap. II and considered in more detail in Ref. 11.

An alternative approach, which has received considerable attention in the literature, is a gradient method[12-14]. Basically, this approach is based on iterative computations using a linear expansion about the previous solution. In Chap. III a first-order gradient method is discussed. The method is explained in detail, and a number of examples are worked.

Based on the results of this chapter, it appears that on-line computation with the gradient method is difficult but feasible. If computations are repeated each time a new measurement is received, the changes in the problem will likely be so small that only a few iterations are required to obtain the new solution. Work is progressing on second-order gradient methods that have better convergence properties when the previous solution is very close to the present solution. Also, if these changes are sufficiently small, it may be possible to utilize a given solution for several time increments, until the additional measurements change the solution enough to justify a new calculation. However, this possibility has not been studied in sufficient detail to draw any conclusions at this time.

Although the gradient method of Chapter III assumes a fixed final time and a terminal cost function that is a function of only the AMM state, the extension of the method to the performance criterion of Eq. (29) has been worked out. The additional computations required are not extensive. The modifications to the procedure of Chap. III will be discussed in a later memorandum.

The computational procedure for this case is summarized in the flow chart of Fig. 2. Target estimation and prediction are performed as in the previous section. The present AMM state is obtained as $\underline{x}_A^{(P)}$ using whatever data processing methods are necessary Based on these data and the previous solution, which has been stored, a new solution is computed by the gradient procedure The optimal guidance command for the particular time is transmitted directly to the AMM. The entire solution is retained, both for use the next time the gradient procedure is applied and for communication to the AMM if no further data are received.

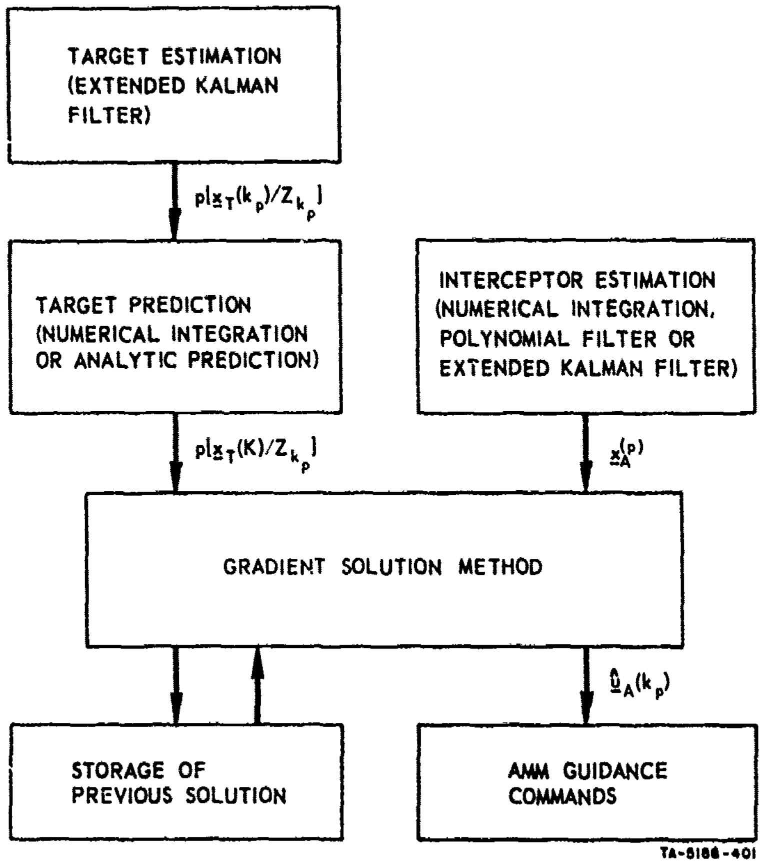

FIG. 2 FLOW CHART FOR SOLUTION AFTER LAUNCH

C. Interception of an MRV

If the target is a maneuvering reentry vehicle (MRV), the problem can be formulated as in Sec. B, the only difference being that the MRV is capable of altering its course during flight. The dynamic equations of the target thus must be modified to read

$$\dot{\underline{x}}_T = \underline{f}_T(\underline{x}_T, \underline{u}_T, t) + \underline{w}_T \tag{33}$$

where $\underline{u}_T$ is the vector of maneuvering controls and where $\underline{f}_T$ now includes the maneuvering forces.

The presence of controllable maneuvering forces greatly increases the chances of the target escaping interception by the AMM. The degree to which these maneuvers are assumed to be evasive determines what problem formulation is appropriate. Under one viewpoint, it is assumed that

the target processes data about the AMM trajectory and then takes direct evasive action. The problem then is a *stochastic differential game*. Using the notation of Sec. B and Eq. (33), the formulation is the following.

Given:

(1) Dynamic Equations for the MRV and AMM

$$\dot{\underline{x}}_T = \underline{f}_T(\underline{x}_T, \underline{u}_T, t) + \underline{w}_T$$

$$\dot{\underline{x}}_A \quad \underline{f}_A(\underline{x}_A \; \underline{u}_A, t) + \underline{w}_A$$

(2) Measurement Equations for the AMM

$$\underline{z}_{TA} = \underline{h}_{TA}(\underline{x}_T, t) + \underline{v}_{TA}$$

$$\underline{z}_{AA} = \underline{h}_{AA}(\underline{x}_A, t) + \underline{v}_{AA}$$

(3) Measurement Equations for the MRV

$$\underline{z}_{TT} = \underline{h}_{TT}(\underline{x}_T, t) + \underline{v}_{TT}$$

$$\underline{z}_{AT} = \underline{h}_{AT}(\underline{x}_A, t) + \underline{v}_{AT}$$

(4) Statistics on the random forcing functions, the measurement noises, and the *a priori* states of the MRV and AMM

$$p(\underline{w}_{TA}),\; p(\underline{w}_{TT}),\; p(\underline{v}_{TA}),\; p(\underline{v}_{TT}),\; p_0[\underline{x}_T(t_0)]$$

$$p(\underline{w}_{AA}),\; p(\underline{w}_{AT}),\; p(\underline{v}_{AA}),\; p(\underline{v}_{AT}),\; p_0[\underline{x}_A(t_0)]$$

(5) All past measurements of the MRV and AMM available to the AMM

$$\underline{z}_{TA}(\sigma), t_0 \leq \sigma \leq t_p$$

$$\underline{z}_{AA}(\sigma), t_0 \leq \sigma \leq t_p$$

(6) All past measurements of the MRV and AMM available to the MRV

$$\underline{z}_{TT}(\sigma), t_0 \leq \sigma \leq t_p$$

$$\underline{z}_{AT}(\sigma), t_0 \leq \sigma \leq t_p$$

(7) A performance criterion

$$J = E\left\{\int_{t_l}^{t_f} l_A[\underline{x}_A(\sigma), \underline{u}_A(\sigma), \sigma]\, d\sigma - \int_{t_0}^{t_f} l_T[\underline{x}_T(\sigma), \underline{u}_T(\sigma), \sigma]\, d\sigma + \psi[\underline{x}_A(t_f), \underline{x}_T(t_f), t_f]\right\}$$

(8) Constraints on the MRV and AMM

$$\underline{x}_T \in X_T$$

$$\underline{u}_T \in U_T$$

$$\underline{x}_A \in X_A$$

$$\underline{u}_A \in U_A$$

Find:

A launch time, t_l, an intercept time, t_f, and an AMM control policy $\underline{u}_A(\cdot)$, $t_l \leq \sigma \leq t_f$, such that the performance criterion in (5) is minimized over all MRV control policies, $\underline{u}_T(\sigma)$, $t_0 \leq \sigma \leq t_f$, where the expectation is conditioned on all past measurements and *a priori* statistics available to the AMM, and where the constraints in (8), the dynamic equations in (1), and the measurement equations in (2) are all satisfied.

The major difference between this problem and that of Sec. B is that the control policy selected by the AMM is influenced by the possible strategies of the MRV. Specifically, the AMM is attempting to minimize the performance criterion based on the measurements it receives, while the MRV is simultaneously trying to maximize this criterion based on its own measurements. The criterion thus must reflect the objectives of both sides. It may then be appropriate to consider in the terminal cost function the damage that the MRV can inflict if it detonates its warhead just before being intercepted. In addition, there may be constraints that the MRV has to hit a particular defended area; this is an extremely important consideration in hard-point defense. Finally, it may be desirable to assign a cost to the MRV for performing certain types of maneuvers; a term in (5) is provided for this purpose.

Relatively little work has been done on the general case of this problem.[4] A summary of results obtained to date will appear in a later memorandum.

This formulation is unrealistic in that it assumes that the MRV is capable of making measurements of the AMM and processing this data on-board. This capability appears to be beyond the capacity of MRV threats in the foreseeable future.

A simpler formulation that has received considerable attention in the literature is that of the deterministic differential game[15-17]. In this problem it is assumed that both the target and the AMM have complete knowledge of each other's state. Again, the assumption that the MRV knows the state of the AMM is unrealistic. It is shown in Chapter IV that the use of this assumption can lead to results that are unduly pessimistic for the defense.

A problem of considerable practical interest is that in which the AMM receives measurements of the MRV state, but the MRV is denied any information about the AMM trajectory. In Chap. IV a simplified version of this case is studied extensively using the concepts of ordinary game theory. Although the problem studied is still far from a realistic model of an engagement, the results are the first obtained using this approach. This work is also noteworthy in that the use of multiple AMM's to intercept a single MRV is allowed and the defense of a hardened site is taken into account.

Further work is currently in progress on the last-mentioned approach and other gaming approaches to MRV intercept[18]. An extensive study of the problem of estimating the state of an MRV appears in Ref. 1. A report on all of this work will appear midway through this year.

D) Contents of the Remainder of this Report

The remainder of this report is divided into three chapters. Each chapter discusses a computer program for performing the calculations required in one of the problem formulations discussed in the previous section. Each chapter is self-contained for the benefit of a reader interested in one or more, but not necessarily all of these programs.

Chapter II discusses a forward dynamic programming program for use in the prelaunch calculation. As discussed in Sec. B-3, this program is to be run off-line, and the results are to be retrieved in real time. Programs for the on-line recovery are also described. The basic program is written for general AMM dynamics. There is also considerable flexibility in the choice of both the integral cost incurred during the flight of the AMM and the terminal cost. Results for particular cases are discussed in this chapter.

Chapter III discusses a first-order gradient method program for guidance after launch. Again, there is considerable flexibility in the specification of AMM dynamics and the performance criterion. Results for a number of specific cases are presented.

Finally, Chap. IV summarizes the results that have been obtained using a game-theoretic approach for intercepting an MRV that does not receive information about the AMM trajectory. A comparison of these results with those for a differential-game approach is given. Also, hard-point defense and the use of multiple AMM's for the interception of a single MRV are discussed.

REFERENCES

1. R. E. Larson, R. M. Dressler, R. S. Ratner, B. L. Ho, "Applications of the Extended Kalman Filter to BRV and MRV Trajectory Estimation," Final Report, Contract DA-01-021-AMC-90006(Y), SRI Project 5188-203, Stanford Research Institute, Menlo Park, California (December 1967).

2. R. E. Larson, "Optimum Guidance Laws for an AMM," Preliminary Report, Contract DA-01-021-AMC-90006(Y), SRI Project 5188-7, Stanford Research Institute, Menlo Park, California (January 1966).

3. L. Meier, "Combined Optimum Control and Estimation," *Proc. Third Allerton Conference*, University of Illinois, Urbana, Ill. (October 1965).

4. I. B. Rhodes, PhD Thesis, Department of Electrical Engineering, Stanford University, Stanford, California (February 1968).

5. R. E. Larson, R. M. Dressler, and R. S. Ratner, "Application of the Extended Kalman Filter to Ballistic Trajectory Estimation," Final Report, Contract DA-01-021-AMC-90006 (Y) SRI Project 5188 103, Stanford Research Institute, Menlo Park, California (January 1967).

6. R. S. Ratner, "Noise Model of NIKE-X MSR for Implementation in an Extended Kalman Filter," (U) Memorandum 22, Contract DA-01-021-AMC-90006(Y), SRI Project 5188-103, Stanford Research Institute, Menlo Park, California (February 1967) (SECRET)

7. R. E. Larson, *State Increment Dynamic Programming* (American Elsevier Publishing Company, New York, New York 1968).

8. R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming* (Princeton University Press, Princeton, New Jersey 1962).

9. R. E. Larson, "Partitioning of the Combined Optimal Control and Estimation Problem for the Nike-X System," Memorandum 20, Contract DA-01-021-AMC-90006(Y), SRI Project 5188-103, Stanford Research Institute, Menlo Park, California (August 1966).

10. R. E. Kalman, "New Methods and Results in Linear Prediction and Filtering Theory," RIAS TR 61-1, Research Institute for Advanced Studies, Baltimore, Maryland (January 1961).

11. W. G. Keckler, "Optimization About a Nominal Trajectory Via Dynamic Programming," Engineer's Thesis, Department of Electrical Engineering, Stanford University, Stanford, California (in preparation).

12. J. V. Breakwell, "The Optimization of Trajectories," *J. SIAM*, Vol. 7, No. 2, pp. 215-247 (June 1959).

13. H. J. Kelley, "Method of Gradients," Chapter 6, pp. 205-224 of *Optimization Techniques*, G. Leitmann, ed., (Academic Press, New York, New York 1962).

14. A. E. Bryson and W. F. Denham, "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Trans. ASME, J. Appl. Mechs.*, Vol. 29, No. 2, pp. 247-259 (June 1962).

15. R. Isaacs, *Differential Games* (John Wiley and Sons, Inc., New York, New York 1965).

16. Y. C. Ho, A. E. Bryson, and S. Baron, " Differential Games and Optimal Pursuit-Evasion Strategies," *IEEE Trans. Automatic Control*, Vol. AC-10, No. 4, pp. 385-389 (October 1965).

17. P. Meschler, "On a Goal Keeping Differential Game," *IEEE Trans. Automatic Control*, Vol. AC-12, No. 1, pp. 15-21 (February 1967).

18. R. E. Larson, "Preliminary Survey of Studies to be Performed for the NIKE-X Project Office by the Information and Control Laboratory," SRI Internal Memorandum to P. E. Merritt (July 19, 1967).

PREVIOUS PAGE IS BLANK

Chapter II

APPLICATION OF DYNAMIC PROGRAMMING TO THE PRELAUNCH CALCULATION

A. Introduction

In a missile defense system the initial trajectory along which the antimissile missile (AMM) is launched is a critical factor in system performance. As discussed in the previous chapter, the determination of these trajectories is made in conjunction with the launch doctrine as part of the prelaunch calculation. This chapter describes a dynamic programming computational procedure that provides an extremely useful tool for obtaining these trajectories. A computer program based on this procedure has been written. This program can be used either to precompute results for later retrieval in an on-line system or else to evaluate off-line alternative launch doctrines.[1,2]

The problem considered in this chapter is the following:

> For an AMM with known dynamics located at a given launch site, find the minimum-time trajectory from the launch site to any reachable point in space for all feasible values of the velocity-vector angle.

The criterion of minimum-time-to-intercept maximizes the altitude of intercept, and hence provides the greatest protection to the defended area. The angle-of-velocity vector is considered because of the importance of the crossing angle—*i.e.* the angle at intercept between the velocity vector of the AMM and the velocity vector of the reentry vehicle (RV)—in the ability of the AMM to compensate for errors in predicted RV position. Such errors are always present because of uncertainty in the estimates of present RV position, velocity, and dynamics.

The minimum time to reach all possible intercept points at all feasible velocity-vector angles is required in order to determine, on the basis of statistics of predicted RV position, whether or not to launch at the present time (Chap. I). If the decision is made to launch, the intercept point is specified, and the corresponding AMM control sequence

and trajectory can be retrieved. The criterion used to make this decision is specified as part of the launch doctrine. It should be emphasized that the same values of minimum time can be used in conjunction with any launch doctrine and for any statistics of predicted RV position; hence, these calculations need be made only once for given AMM dynamics.

Dynamic programming provides an effective method for obtaining the required information. Extremely general AMM dynamics can be handled; and, if desired, a performance criterion other than minimum time can be used. Constraints introduce no difficulties, and they actually reduce the computational burden by decreasing the size of the region that must be searched. The direct-search procedure guarantees that an absolute optimum is obtained. The minimum-time trajectories to all reachable intercept points for all feasible velocity-vector angles are determined in a single calculation, and the corresponding control sequences can be recovered from these results with no additional computations. This latter property is a direct consequence of the use of forward dynamic programming[3], and it is analogous to the feedback control property of the conventional dynamic programming procedure[4,5]. However, it should be noted that once the AMM is launched, the results of this program do *not* necessarily specify the minimum-time trajectory to any other intercept point and velocity-vector angle. If later information about the predicted RV position dictates a change in these quantities, then the use of the gradient method[6-8], or some other technique based on perturbations from a nominal trajectory, should be used.

In Sec. B of this chapter the basic relations of dynamic programming are derived, and their significance is discussed. Section C describes forward dynamic programming and illustrates its application to a simple example. Section D formulates the problem of finding the control sequence that reaches a specified intercept position and velocity angle while minimizing an integral performance criterion (see Chap. I). The performance criterion considered in this section is minimum time, but other cost functions can be utilized. Section E describes the computational procedure used in the computer program which obtains the desired control sequences. In Sec. F representative results from the computer program are presented.

Find:

The control sequence $\underline{u}(0), \ldots, \underline{u}(K)$ such that J in Eq. (2) is minimized subject to the system equation (1), the constraint equations (3) and (4), and the initial condition (5).

2. Derivation of the Basic Iterative Functional Equation

The dynamic programming solution to the above problem is obtained by using an iterative functional equation that determines the optimal control for any admissible state at any stage. This equation follows immediately from Bellman's principle of optimality.[1-3] However, in the interest of clarity, a direct derivation will be given, and the principle of optimality will then be interpreted in terms of this equation.

The first step in the derivation is to define the minimum cost function for all $\underline{x} \in X$ and all k, $k = 0, 1 \ldots, K$, as

$$I(\underline{x},k) = \min_{\underline{u}(j),\, j=k,\ldots,K} \sum_{j=k}^{K} l[\underline{x}(j),\underline{u}(j),j] \quad , \tag{6}$$

where

$$\underline{x}(k) = \underline{x} \quad .$$

The summation is then split into two parts, the term evaluated for $j = k$, and the summation over $j = k + 1$ to $j = K$. The minimization is similarly split into two parts. The resulting equation is

$$I(\underline{x},k) = \min_{\underline{u}(k)} \min_{\substack{\underline{u}(j) \\ j=k+1,\ldots,K}} \left\{ l[\underline{x},\underline{u}(k),k] + \sum_{j=k+1}^{K} l[\underline{x}(j),\underline{u}(j),j] \right\} \quad . \tag{7}$$

The first term in brackets in Eq. (7) is not affected by the second minimization. Thus, the equation becomes

$$I(\underline{x},k) = \min_{\underline{u}(k)} l[\underline{x},\underline{u}(k),k] + \min_{\substack{\underline{u}(j) \\ j=k+1,\ldots,K}} \sum_{j=k+1}^{K} l[\underline{x}(j),\underline{u}(j),j] \quad , \tag{8}$$

The second term in brackets in Eq. (8) is exactly analogous to the definition in Eq. (6), where the argument of I is $\underline{g}[\underline{x},\underline{u}(k),k]$, $k + 1$. Abbreviating $\underline{u}(k)$ as $\underline{u}$, the iterative functional equation becomes

$$I(\underline{x},k) = \min_{\underline{u}} \{ l(\underline{x},\underline{u},k) + I[\underline{g}(\underline{x},\underline{u},k),\ k+1] \} \quad . \tag{9}$$

This equation is a mathematical statement of Bellman's principle of optimality.[1-3] It states that the minimum cost for state $\underline{x}$ at stage k is found by choosing the control that minimizes the sum of the cost to be paid at the present stage and the minimum cost in going to the end from the state at stage $k + 1$ which results from applying this control. The optimal control at state $\underline{x}$ and stage k, denoted as $\hat{\underline{u}}(\underline{x},k)$, is directly obtained as the value of $\underline{u}$ for which the minimum in Eq. (9) is attained.

Since Eq. (9) determines $I(\underline{x},k)$ and $\hat{\underline{u}}(\underline{x},k)$ in terms of $I(\underline{x},k + 1)$, it must be solved backward in k. As a terminal boundary condition,

$$I(\underline{x},K) = \min_{\underline{u}} [l(\underline{x},\underline{u},K)] \quad . \tag{10}$$

The optimization over a sequence of controls is thus reduced to a sequence of optimizations over a single control vector.

3. The Standard Computational Algorithm

In the standard method for solving Eq. (9), each state variable x_i, $i = 1, 2, \ldots, n$, is quantized to N_i levels and each control variable u_j, $j = 1, 2, \ldots, m$, is quantized to M_j levels.

Initially, $I(\underline{x},K)$ is found for all quantized states $\underline{x} \in X$ by evaluating $l(\underline{x},\underline{u},K)$ for each quantized control $\underline{u} \in U$ and choosing the minimum value by a direct comparison. The optimal control, $\hat{\underline{u}}(\underline{x},K)$, is the value of $\underline{u}$ that minimizes $l(\underline{x},\underline{u},K)$. In many problems no control is applied at the final stage K; in this case $I(\underline{x},K)$ is evaluated directly as $l(\underline{x},K)$.

Next, at $k = K - 1$, for each quantized state $\underline{x} \in X$, each quantized control $\underline{u} \in U$ is applied, and the next state $\underline{g}(\underline{x},\underline{u},K)$, is computed. The minimum cost at the next state, $I[\underline{g}(\underline{x},\underline{u},k),K]$, is found by interpolation using the values of $I(\underline{x},K)$ at quantized states. The quantity $l(\underline{x},\underline{u},K - 1)$ is computed directly. The sums of these quantities for each quantized control are then compared, and the minimum value is stored as $I(\underline{x},K - 1)$. The optimal control, $\hat{\underline{u}}(\underline{x},K - 1)$, is stored as the value of $\underline{u}$ for which the minimum is attained. The procedure continues in this manner, with $I(\underline{x},k)$ and $\hat{\underline{u}}(\underline{x},k)$ being computed in terms of $I(\underline{x},k + 1)$, until $k = 0$ is reached.

C. Forward Dynamic Programming

If the minimum cost function is redefined to be the minimum cost to reach a given state and stage from the initial state, an iterative equation analogous to Eq. (10) can be derived. In this case the calculations proceed forward in k rather than backward; hence, the term *forward dynamic programming* is used to describe the computational procedure for this case.

The iterative equation is derived by defining $I'(\underline{x},k)$ as the minimum cost to reach state $\underline{x}$ at stage k from the initial state. Formally,

$$I'(\underline{x},k) = \min_{\underline{u}(0),\underline{u}(1)\ldots\underline{u}(k-1)} \sum_{j=0}^{k-1} l[\underline{x}(j),\underline{u}(j),j] \tag{11}$$

where

$$\underline{x} = \underline{g}[\underline{x}(k-1),\underline{u}(k-1),k-1] \quad .$$

If the inverse functional to $\underline{g}$ is defined as $\underline{h}$, so that

$$\underline{g}\{\underline{h}[\underline{x},\underline{u}(k-1),k-1],\underline{u}(k-1),k-1\} = \underline{x} \quad , \tag{12}$$

then the iterative equation becomes

$$I'(\underline{x},k) = \min_{\underline{u}(k-1)} (l\{\underline{h}[\underline{x},\underline{u}(k-1),k-1],\underline{u}(k-1).k-1\}$$

$$+ I'\{\underline{h}[\underline{x},\underline{u}(k-1),k-1],k-1\}) \quad . \tag{13}$$

As a boundary condition for this equation, $I'(\underline{x},0)$ is specified. If, as is often the case, the initial state is fixed at one particular value, $I'(\underline{x},0)$ is set to zero for this state and no other initial state is considered admissible.

One method of carrying out the computations in Eq. (13) is to use a procedure exactly analogous to that of the previous section. In this case, at a particular quantized $\underline{x}$ and k, each quantized control $\underline{u}(k-1)$ is applied. the corresponding previous state is found as $\underline{h}[\underline{x},\underline{u}(k-1),k-1]$; the minimum cost for this state, $I(\underline{h},k-1)$, is found, using interpolation if necessary; $l[\underline{h},\underline{u}(k-1),k-1]$ is evaluated directly; and the optimal control and minimum cost at $\underline{x}$, k are obtained by comparing the quantities in brackets in Eq. (13) for all quantized controls $\underline{u}(k-1)$. This

procedure is straightforward, but it does require computation of the inverse functional $\underline{h}$; this can be difficult if $\underline{g}$ is a general nonlinear time-varying functional.

A procedure that overcomes this difficulty is the following: At each quantized state $\underline{x}(k-1)$ where $I'[\underline{x}(k-1),k-1]$ has just been computed, each quantized admissible control is applied; for each corresponding next state, $\underline{x}(k) = \underline{g}[\underline{x}(k-1),\underline{u}(k-1),k-1]$, a check is made to see if it has been the next state for any control applied at previous values of $\underline{x}(k-1)$; if it has not previously been a next state, then the quantity in brackets in Eq. (13) is stored as the tentative minimum cost at that point; if it has been, then the quantity in brackets in Eq. (13) is compared with the tentative minimum cost already computed at that point, and, if it is less, this minimum cost and optimal control replace the values stored there. If the next state does not fall exactly at a quantized state, then interpolation is required. This procedure continues until the quantized admissible controls have been applied at every quantized state $\underline{x}(k-1)$. The tentative minimum costs and optimal controls at each

$$\underline{x}(k) = \underline{g}[\underline{x}(k-1),\underline{u}(k-1),k-1]$$

are then the true minimum costs and optimal controls at these points.

Since this procedure is to be used to solve the prelaunch problem, a simple one-dimensional example will be worked out. The system equation for this problem is

$$x(k+1) = x(k) + u(k) \quad . \tag{14}$$

The performance criterion is

$$J = \sum_{j=0}^{4} [x^2(j) + u^2(j)] \quad . \tag{15}$$

The constraints are

$$-1 \leq u(k) \leq 1 \tag{16}$$

and

$$0 \leq x(k) \leq 2 \quad . \tag{17}$$

The initial state is known to be

$$x(0) = 2 \quad . \tag{18}$$

The quantization increments are taken to be $\Delta x = 1$ and $\Delta u = 1$. The first step in the procedure is to apply the three quantized admissible controls, $u = -1$, $u = 0$, and $u = +1$, at the initial state, $x(0) = 2$. The minimum cost and optimal control for admissible states at $x(1)$ are found without need for comparison. These values are shown in Fig. 1. Note that the optimal control determines the preceding state, not the succeeding state.

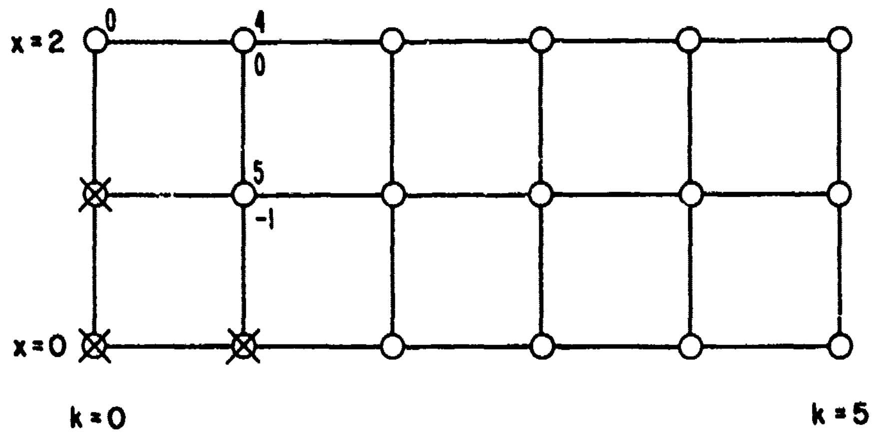

FIG. 1 FIRST STEP IN FORWARD DYNAMIC PROGRAMMING PROCEDURE

In going from $I'(x,1)$ to $I'(x,2)$ a comparison is required in some of the calculations. If the controls are first applied at $x(1) = 2$, the tentative minimum costs and optimal controls are as shown in Fig. 2. Asterisks are placed beside these values to show that they are tentative.

When the controls at $x(1) = 1$ are applied, the states $x(2) = 2$ and $x(2) = 1$ are possible next states. The minimum costs coming from $x(1) = 1$ are compared with the values already there in Fig. 2; in both cases, less cost is obtained when $x(1) = 1$. The complete results at $k = 2$ are thus as shown in Fig. 3.

This procedure continues until minimum cost and optimal control have been computed at all quantized values of x and k. The results are shown in Fig. 4.

These results can be used in the following ways. First, suppose that the final state is constrained to be $x(5) = 2$. In this case, the optimal trajectory can be found by tracing back along the trajectory starting at $x(5) = 2$. The result is shown in Fig. 5.

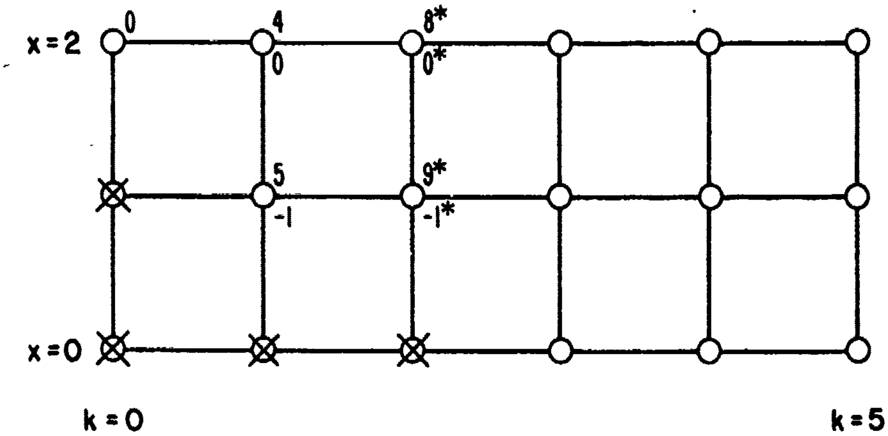

FIG. 2 TENTATIVE MINIMUM COSTS AT k = 2

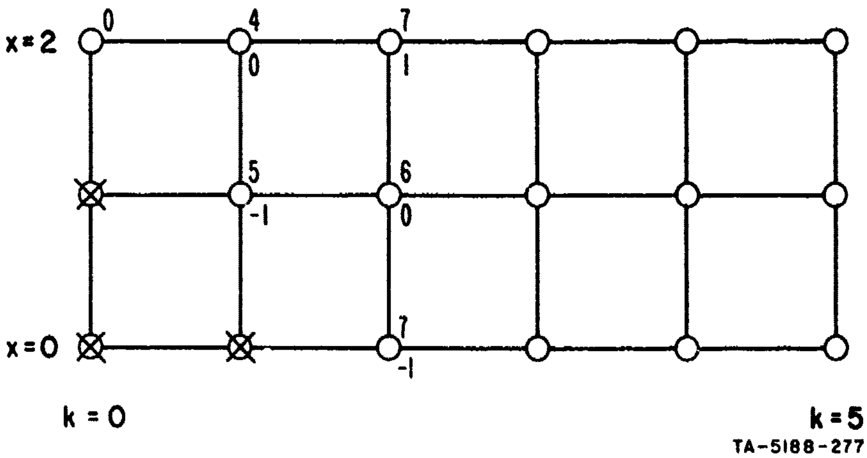

FIG. 3 COMPLETE RESULTS AT k = 2

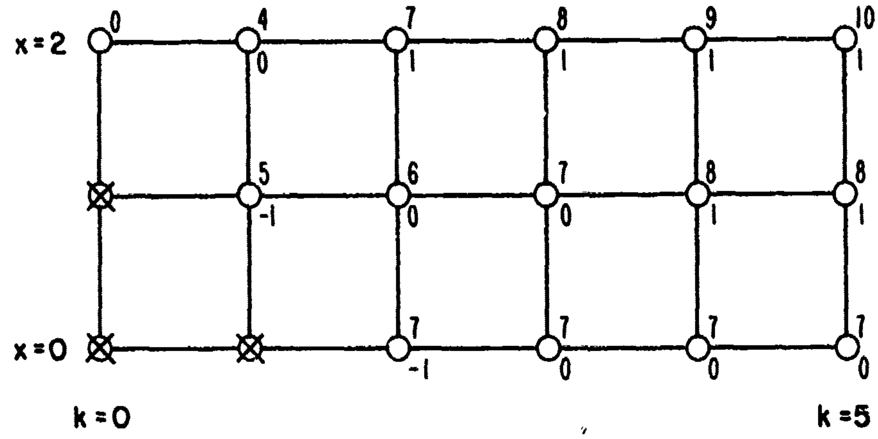

FIG. 4 COMPLETE RESULTS FOR FORWARD DYNAMIC PROGRAMMING EXAMPLE

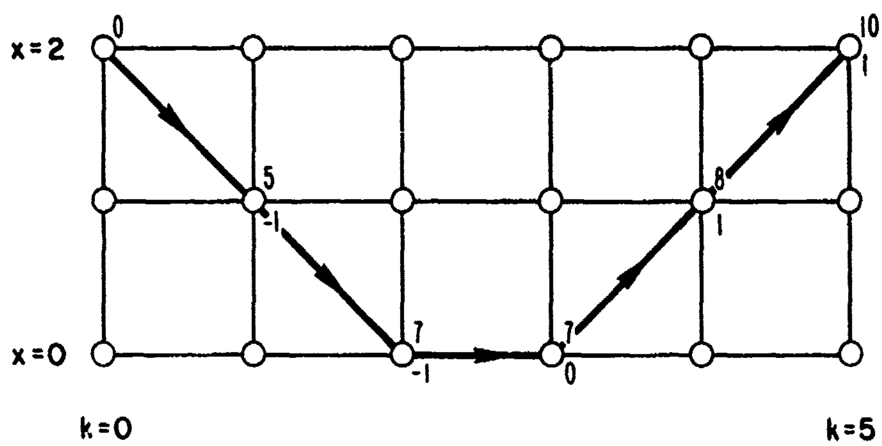

FIG. 5 OPTIMAL TRAJECTORY IF FINAL STATE MUST BE x = 2

Next, suppose that the final state can be anywhere in the region $0 \leq x \leq 2$. In this case a second search is made over the minimum costs at the quantized final states, and the final state is taken to be the one for which the minimum cost is least. In this example, the minimum cost is $I' = 7$ at $x(5) = 0$. The optimal trajectory is then as shown in Fig. 6.

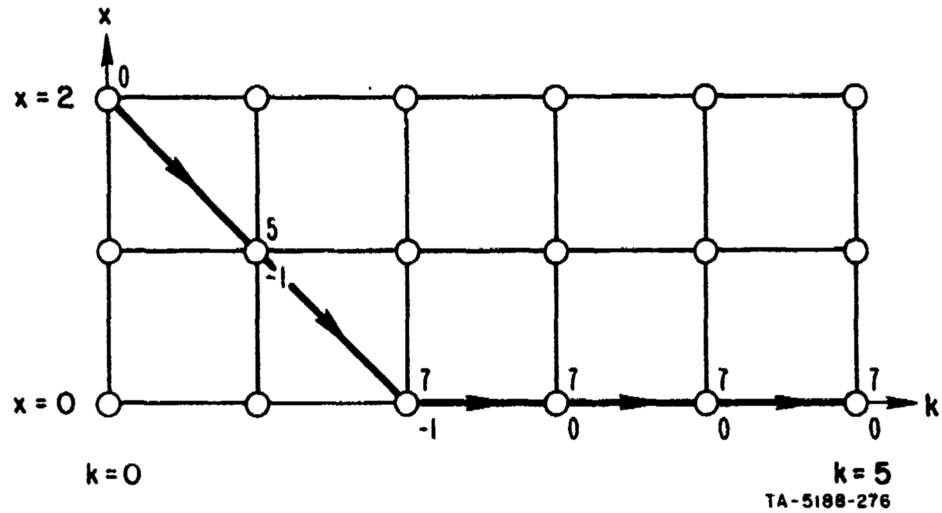

FIG. 6 OPTIMAL TRAJECTORY IF FINAL STATE IS NOT CONSTRAINED

Another possibility is that the final state can be anywhere in the region $0 \leq x \leq 2$, but there exists a terminal cost function

$$\psi[x(K),u(K)] = l[x(K),u(K),K] \quad . \tag{19}$$

In particular, suppose that the example has a terminal cost of the form

$$\psi[x(5),u(5)] = 2.5[x(5) - 2]^2 \quad . \tag{20}$$

Table 1 shows the results. When the terminal cost is considered, termination in state $x = 2$ minimizes the cost of operating the process. The resulting optimal trajectory is shown on Fig. 5.

This computational procedure can be extended to more complex problems. If the next state,

$$\underline{x} = g[\underline{x}(k - \underline{u}),u(k - 1),k - 1] \quad ,$$

does not occur exactly at quantized values, some interpolation procedure must be used. The simplest technique is to associate the next state with

Table I

VALUES OF TERMINAL COST FUNCTION AND THE TOTAL COST OF TERMINATING IN EACH QUANTIZED STATE

X \	J	ψ	TOTAL COST
0	10	0	10
1	8	2.5	10.5
2	7	10	17

the nearest quantized state for purposes of cost comparison. The optimal cost $I(\underline{x}, k)$ will be the minimum cost chosen from all states for which $\underline{x}$ is the nearest quantized value. The optimal control and minimum cost and the corresponding state need not be a quantized value. This eliminates the need for interpolations in reconstructing optimal trajectories after computations have been completed—an especially useful property if the optimal trajectories are physical missile trajectories.

D. Formulation of the Minimum-Time Problem

The purpose of the solution is to determine the minimum-time trajectory from the missile launch site to any point in three-dimensional space within the range of the missile. However, it is apparent that minimum-time missile trajectories always stay in a plane determined by the intercept point, the launch point, and the local vertical at the missile site. Thus, solution of the problem of minimum-time trajectories to all points in a vertical plane yields the solution to the three-dimensional problem. The kinematic equations for this reduced space are the following:

$$\ddot{y} = 1/m[T \cos (\varphi + \alpha) - D \cos \varphi - L \sin \varphi]$$

$$\ddot{z} = 1/m[T \sin (\varphi + \alpha) - D \sin \varphi + L \cos \varphi] - g \qquad (21)$$

where

T = Thrust of AMM (lbs)

D = Drag along flight path (lbs)

m = Mass of vehicle (slugs)

L = Control force applied perpendicular to flight path (lbs)

φ = Flight path angle

α = Angle of attack

g = Gravitational acceleration (ft/sec^2)

y - Horizontal distance from the launch point (ft)

z = Vertical distance from the launch point (ft).

Figure 7 shows the missile geometry.

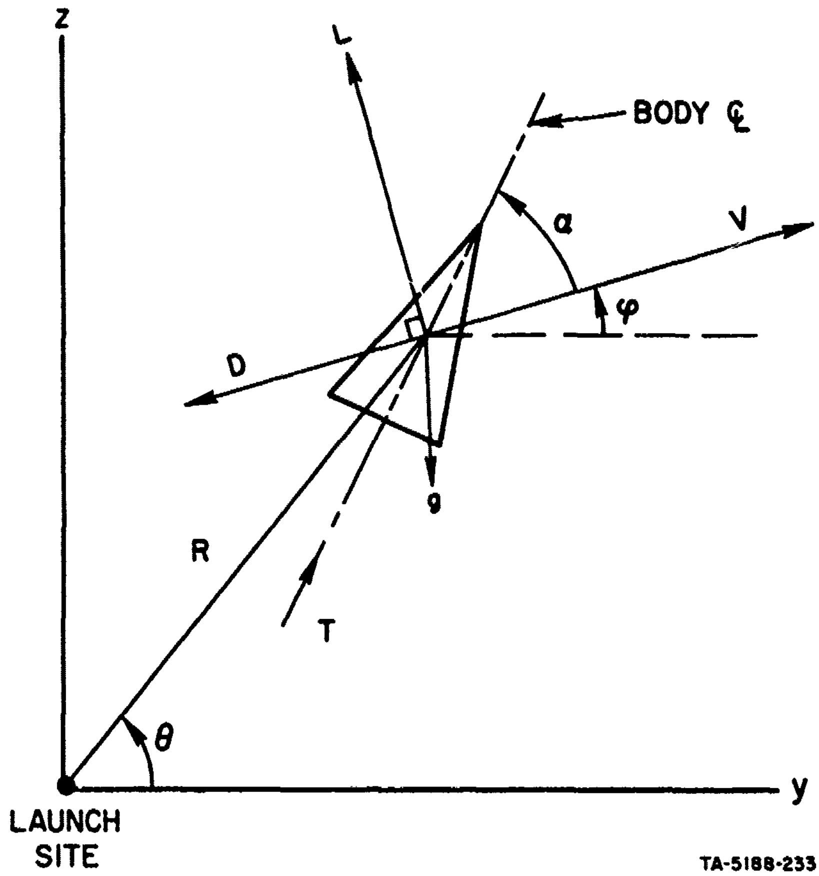

FIG. 7 MISSILE GEOMETRY

The drag is computed as

$$D = \frac{1}{2} \rho V^2 S(C_{D_o} + C_{D_i}) \tag{22}$$

where

ρ = Atmospheric density (slugs/ft^3)

V = Speed (ft/sec)

S Reference area of the missile (ft^2)

C_{D_o} = Zero-lift drag coefficient

C_{D_i} = Coefficient of drag induced by maneuver.

The quantity L is the commanded lift force perpendicular to the flight path. It can be expressed:

$$L = \frac{1}{2} \rho V^2 S C_L \tag{23}$$

where

C_L = Lift coefficient .

Thus, once a desired control has been chosen, the desired lift coefficient is computed as soon as the other parameters in the equation are known.

There are numerical difficulties because of the nonlinearity of the parameters. The atmospheric density is a function of altitude which is best determined by a table-lookup procedure. The lift and drag coefficients are functions of the Mach number (which in turn depends on the altitude and velocity) and the angle of attack required to implement the commanded lateral acceleration. All of these relations are found by table lookup and then assumed to be constant for a time that is short with respect to the rates of change of the parameters. However, these variations in parameters do preclude getting closed-form solutions to the minimum-time problem.

The state of the missile is completely known if the position and velocity are known. Since the missile is restricted to motion in a plane there are two components of both position and velocity—a total of four state variables. If the assumption is made that acceleration is constant during the interval (Δt) in which the missile parameters are constant, the following integration formula is used:

$$\begin{aligned} \dot{y}(t + \Delta t) &= \dot{y}(t) + \ddot{y}(t)\Delta t \\ \dot{z}(t + \Delta t) &= \dot{z}(t) + \ddot{z}(t)\Delta t \end{aligned} \tag{24}$$

and

$$y(t + \Delta t) = y(t) + \frac{\Delta t}{2} [\dot{y}(t) + \dot{y}(t + \Delta t)]$$

$$z(t + \Delta t) = z(t) + \frac{\Delta t}{2} [\dot{z}(t) + \dot{z}(t + \Delta t)] \quad .$$

A program could be written using Eqs. (21), (22), (23), and (24) to define the state space and the missile dynamics. However, this would

yield a problem consisting of four state variables and time. Since time is the stage variable, the program would have five quantized variables. Fortunately, the dynamics of the AMM are such that certain approximations can be made. One of these is that the distance from the launch point to missile is monotonically increasing with time. Since minimum-time paths are of prime consideration, it is not likely that such paths would double back. The radial distance from the launch site (R) can thus be used as a stage variable. The angle of vector from the launch point to the AMM (θ) can be used as the second state variable of position. Thus, y and z are replaced by R and θ, and time appears in the problem formulation only as the cost criterion, if it appears at all.

The minimum time cost criterion is a function of four state variables—two to specify position and two to specify velocity. However, the relatively low maneuvering capability of the AMM with respect to its range suggests another approximation. The physical characteristics of the AMM reduce the number of feasible trajectories that reach a point specified by R, θ, and φ (the flight path angle) to those for which the cost criterion minimum time is only a weak function of velocity magnitude. Thus, good approximations of minimum time trajectories can be obtained by ignoring the dependence on velocity magnitude and using a space formed by quantization of the variables R, θ, and φ.

The use of these two approximations reduces the dynamic programming formulation with five degrees of freedom (four state variables and the stage variable time) to one with three degrees of freedom (three state variables—one of which is also a stage variable). A problem of this size can be solved on present-day computers.

The AMM maneuvers aerodynamically and any change in direction consumes considerable kinetic energy or velocity. Many maneuvers or an extreme maneuver will result in so much lost velocity that the AMM will clearly not be on a minimum-time trajectory. For long flights the solution using minimum time criterion does not conserve enough velocity in the early stages of flight to yield as short flight times as using a criterion which conserves velocity magnitude. However, for short trajectories in which the velocity lost to drag is not significant, the minimum-time criterion yields shorter flight times than the maximum-velocity criterion. This ambiguity indicates that the approximations described above yield solutions that are optimal in the reduced space, but not necessarily in

the entire position-velocity state space. The maximum velocity criterion yields improved results in those cases in which velocity is important, namely for long flight paths, because it reintroduces velocity into the problem. Thus, the state reduction assumption described above clearly does lead to errors, but experience has shown that a judicious choice of cost criterion keeps this error well within tolerable limits. The length of flights being considered determines the criterion used in the computer program that generates minimum-time paths.

E. The Forward Dynamic Programming Solution

This section describes the manner in which forward dynamic programming was applied to the solution of this problem. Equations (21) to (24) are the basis of solution. The state variables R and θ are each quantized into a set of fixed discrete values that serve as reference points in physical space for the trajectories to be computed. The number of quantization levels of φ at each point (R,θ) in physical space is fixed, but the values of these quantized states are determined during the actual application of the dynamic programming algorithm. The method for doing this is discussed later. The control options available are also limited to a discrete set.

The second version of forward dynamic programming discussed in Sec. C is then used to solve this problem. A number of mathematical operations must be performed at each point in state space. The system equations (21) to (24) are used in the program to determine the next state, because the equations of motion are easier to use in Cartesian coordinates, and conversion from one coordinate system to the other is easily done. At each point in state space (R,θ,φ) one first estimates the time necessary to reach the quantization level in R (at $t + \Delta t$—call it R') at the velocity of R, θ, and φ. The other quantities necessary for evaluation of Eqs. (21) to (24) are found from table-lookup subroutines. Equation (24) is then evaluated so that $y(t + \Delta t)$ and $z(t + \Delta t)$ are obtained. Then

$$R(t + \Delta t) = \sqrt{y(t + \Delta t)^2 + z(t + \Delta t)^2} \tag{25}$$

is computed. If this is not close enough to R' (within 3 feet), Δt is corrected and Eq. (24) is reapplied with corrected value of Δt. Two or three iterations are usually sufficient.

The next step is a search for the proper quantization level of θ. The θ' which results from the trial trajectory is

$$\theta' = \tan^{-1} \frac{z(t + \Delta t)}{y(t + \Delta t)} \quad . \tag{26}$$

The quantization level of θ for which $|\theta - \theta'|$ is minimized is the quantization number assigned to θ'. Finally, one must determine near which quantization level of φ the trial trajectory terminates. The computer program chooses as many as five trajectories for each R', θ'. These trajectories are chosen to guarantee a minimum spacing $\Delta\varphi$ between each of the trajectories. The trajectory that best meets the cost criterion is chosen for each feasible quantized value of φ.

F. Results

The program described in the previous section has been used to determine the minimum-time trajectories from the launch point to all quantized points in R and θ within the range of an AMM. It was implemented in FORTRAN on an IBM 7090 computer. When trajectories have been computed for a given value of the stage variable R, the data for the preceding value of R is stored on magnetic tape, and the more recent set of results is transferred to the core storage that the older set had occupied. The whole solutio is eventually stored on magnetic tape. A second program has been written which recovers trajectories from the magnetic tape. The terminal point and desired terminal angle of the velocity vector are entered on punched cards and the trajectory that best meets the specified terminal conditions and the corresponding optimal-control history is traced from the launch site to terminal state. Thus, the computer program needs to be run only once for a given set of missile dynamics and constraints to yield a complete solution for all terminal conditions. The recovery program is then used to obtain any number of trajectories.

A typical trajectory obtained by the program is shown in Fig. 8. In this case the terminal position was chosen to be y = 131,000 ft., z = 12,900 ft. The terminal velocity angle was not specified, but instead it was chosen to minimize the time required to reach the point. This time was 24.96 sec, corresponding to a velocity angle of -35.8°.

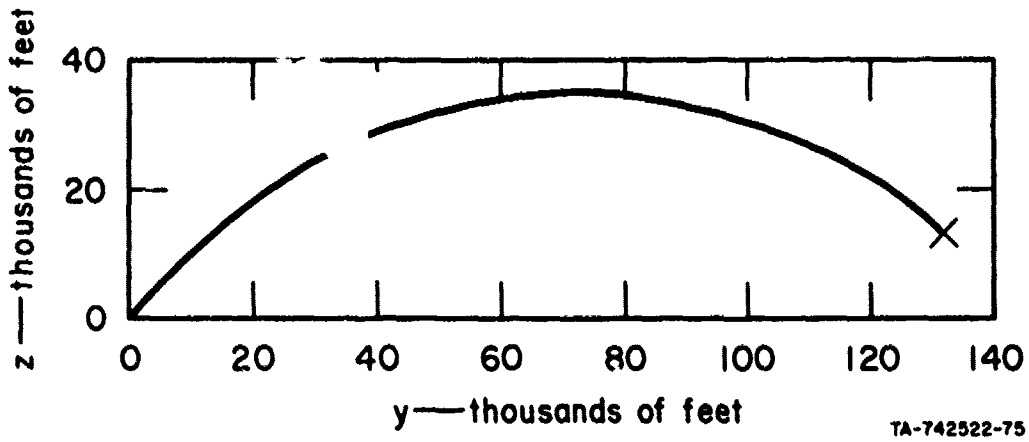

FIG. 8 REPRESENTATIVE MINIMUM-TIME TRAJECTORY TO (y,z) = (131000, 12900)

A number of examples have been run using a realistic set of aerodynamic parameters. The program required less than one hour to compute minimum-time trajectories in the entire operating region of the AMM. The accuracy of the solution has been checked from several points of view. When control histories from this program were used in a routine that integrated the equations of motion much more carefully, the deviations in terminal positions were 0.2 percent of the trajectory lengths. This indicates that the assumptions utilized to integrate the equations of motion do not invalidate the results.

Studies have also been made to determine the deviation of trajectories obtained from the true optimum trajectories. A second optimization scheme using the same ideas described here has been implemented. However, instead of considering a large region as the solution space, this technique considers only a region surrounding some nominal trajectory. The quantizations of the state variables, stage variables, and control can be much finer without resulting in a problem too large to solve. Optimal trajectories from this formulation approximate the continuous-space optimal trajectories much more closely. When optimal trajectories from the original formulation were used as nominal trajectories in this scheme, only two- or three-percent improvements in flight time were noted. Similar improvements were observed when a gradient procedure (see next hapter) was used to generate minimum-time trajectories. These results are sufficiently accurate for use in the prelaunch calculation or for the evaluation of alternative launch doctrines.

G. Conclusions

In this chapter, the application of a dynamic programming computational procedure to the prelaunch calculation in a missile defense system has been discussed. This procedure determines the minimum-time trajectory to any reachable point in space at any achievable velocity-vector angle. This information provides a basis on which to determine whether or not to launch the AMM at a given threat. If a decision to launch is made, the initial control sequence and corresponding trajectory for the AMM are easily obtained from these data. The results for a given set AMM dynamics can be computed in a single run of a computer program and stored for use in either the evaluation of alternative launch doctrines or in an on-line system.

A computer program based on this procedure has been written and tested for a number of different cases. This program used relatively coarse increment sizes and a simplified state description of the AMM. Analysis of these results, by means of a gradient procedure and a dynamic programming procedure using finer increment sizes, showed that these results were sufficiently accurate for this application. The computational requirements of the program are very reasonable, considering that the program needs to be run only once to obtain all minimum-time trajectories needed in the prelaunch calculation.

REFERENCES

1. R. E. Larson, "Optimum Guidance Laws for an AMM," Preliminary Report, Contract DA 01-021-AMC-90006(Y), SRI Project 5188-7, Stanford Research Institute, Menlo Park, California (January 1966).

2. L. Meier III, "Application of Combined Optimization Theory to the NIKE-X System," presented at NIKE-X Control, Guidance and Intercept Symposium, Redstone Arsenal, Alabama, July 25-26, 1967.

3. R. E. Larson, *State Increment Dynamic Programming* (American Elsevier Press, New York, New York,, 1967).

4. R. E. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, New Jersey, 1957).

5. R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming* (Princeton University Press, Princeton, New Jersey, 1962).

6. J. V. Breakwell, "The Optimization of Trajectories," *J. SIAM*, Vol. 7, No. 2, pp. 215-247 (June 1959).

7. H. J. Kelley, "Method of Gradients," in *Optimization Techniques*, G. Leitmann, ed., (Academic Press, New York, New York, 1962).

8. A. E. Bryson and W. F. Denham, "A Steepest Ascent Method for Solving Optimum Programming Problems," *Trans. ASME, J. Appl. Mech.*, Vol. 29, No. 3, pp. 247-259 (June 1962).

Chapter III

APPLICATION OF GRADIENT METHODS TO THE AMM GUIDANCE PROBLEM

A. Introduction

This chapter describes a numerical procedure for computing angle-of-attack commands for an antimissile missile (AMM) to optimize a given performance criterion. This method is particularly well suited to optimizing about a given nominal trajectory. Thus, it can be applied to the guidance problem after an initial trajectory has been obtained in the prelaunch calculation (see Sec. B of Chap. I and Ref. 1). The performance criterion considered consists of terms for the miss distance of the AMM from the target and the crossing angle of the AMM relative to the target trajectory. In addition, the AMM angle of attack, which corresponds to the control variable, is subject to certain constraints. In Sec. B the AMM equations of motion are described, and in Sec. C the problem is formulated.

This optimization problem is equivalent to a classical two-point boundary-value problem. The gradient method, which has been programmed, calculates numerical solutions by the following recursive process:

(1) Forward integration of the equations of motion for the AMM using a given control history

(2) Backward integration of the adjoint equations, which are obtained by a linearization about the trajectory obtained in (1)

(3) Calculation of the gradient of the performance criterion with respect to the control

(4) Determination of a new control history that enables the procedure to iterate toward the optimal solution.

The gradient method, which is described in Sec. D, yields optimal guidance commands (open-loop control histories) and the corresponding optimal trajectory for the AMM.

Numerical results from several illustrative examples are presented in Sec. E.

B. AMM Equations of Motion

In this chapter the computation of optimal guidance histories for an AMM will be considered as a two-dimensional problem. The forces acting upon the AMM are illustrated in Fig. 1, where the y and z axes define the horizontal and vertical directions at the launch site. Thus, the equations of motion for the AMM are given by*

$$\ddot{y} = \frac{1}{m}\left[(-D + T\cos\alpha)\frac{\dot{y}}{V} - (L + T\sin\alpha)\frac{\dot{z}}{V}\right] \qquad (1)$$

$$\ddot{z} = \frac{1}{m}\left[(-D + T\cos\alpha)\frac{\dot{z}}{V} + (L + T\sin\alpha)\frac{\dot{y}}{V}\right] - g \qquad (2)$$

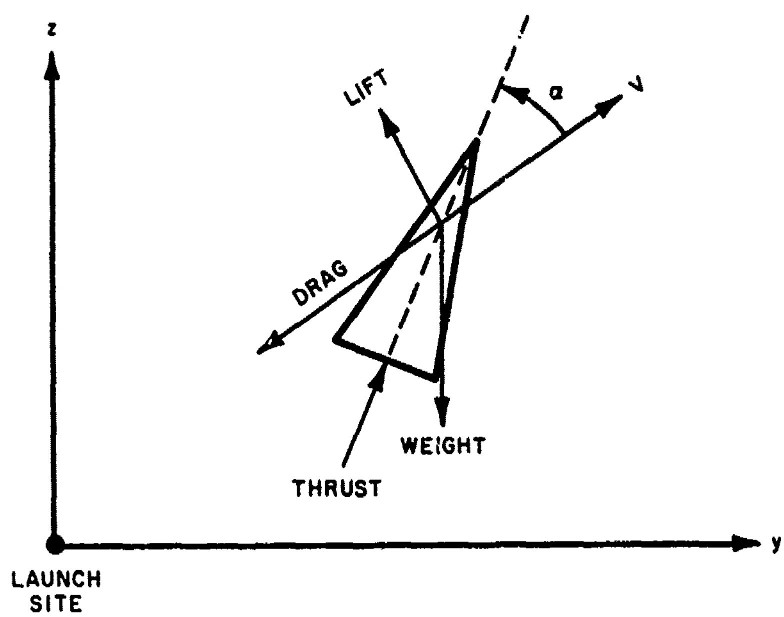

FIG. 1 AMM GEOMETRY

where

$V = \sqrt{\dot{y}^2 + \dot{z}^2}$ (ft/sec)

α Angle of attack - Control variable

T - Thrust (lbs)

* It should be noted that these differential equations are equivalent to the differential equations (21) in Chap. II, since $\dot{y}/V = \cos\varphi$ and $\dot{z}/V = \sin\varphi$ (see Fig. 7 in Chap. II).

m - Mass (slugs)

g - Gravitational acceleration (ft/sec^2)

$$D \quad \text{Drag force} \quad C_D(M,\alpha)S\frac{\rho V^2}{2} \text{ (lbs)} \tag{3}$$

$$L = \text{Lift force} = C_L(M,\alpha)S\frac{\rho V^2}{2} \text{ (lbs)} \tag{4}$$

in which

S = Reference area of the AMM (ft^2)

ρ = Atmospheric density (slugs/ft^3)

$C_D(M,\alpha)$ - Drag coefficient = $C_{D_o}(M) + C_{D_i}(M,\alpha)$

$C_L(M,\alpha)$ Lift coefficient

M = Mach number.

Since short-range AMMs are being considered, the above model assumes a nonrotating and flat earth, with g taken to be a constant. Also ρ is assumed to be a function of the altitude z only. These simplifying assumptions can be removed by including the Coriolis and centrifugal forces acting upon the AMM, and by expressing ρ and g as complex functions of y and z. Furthermore, it has been assumed that the AMM responds instantaneously to commands in α, the angle of attack.

The drag coefficient C_D is the sum of C_{D_o}, the zero-lift drag coefficient, and C_{D_i}, the induced drag coefficient. The drag and lift coefficients are functions of the Mach number M, which depends on the altitude z and the velocity V, and the angle of attack α of the AMM. These relationships are expressed in tabular form. In addition, the Mach number M and the atmospheric density ρ are given in tabular form as functions of the altitude z.

The term L, given by Eq. (4), is the commanded force perpendicular to the flight path of the AMM. The desired lift force for maneuvering is achieved by choosing the control α appropriately. Of course, α will also affect the drag force D as given by Eq. (3).

If the four-dimensional state vector of the AMM is defined as

$$x = \begin{bmatrix} y \\ z \\ \dot{y} \\ \dot{z} \end{bmatrix} , \tag{5}$$

then the differential equations (1) and (2) can be rewritten concisely in state variable form as follows:

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \dot{z} \\ \frac{1}{m}\left[\left(-C_D S \frac{\rho}{2} V^2 + T \cos \alpha\right)\frac{\dot{y}}{V} - \left(C_L S \frac{\rho}{2} V^2 + T \sin \alpha\right)\frac{\dot{z}}{V}\right] \\ \frac{1}{m}\left[\left(-C_D S \frac{\rho}{2} V^2 + T \cos \alpha\right)\frac{\dot{z}}{V} + \left(C_L S \frac{\rho}{2} V^2 + T \sin \alpha\right)\frac{\dot{y}}{V}\right] - g \end{bmatrix} = f(x,\alpha,t) \quad . \tag{6}$$

C. Problem Formulation

The AMM guidance problem being studied can be formulated as follows.

Find:

$$\text{a control history } \alpha(t) \in \Omega(x,t) \qquad \text{for } t_o \leq t \leq t_f \quad , \tag{7}$$

that minimizes the deterministic performance criterion (cost function)

$$J[x(t_f)] \tag{8}$$

subject to the constraint that the state $x(t)$ satisfies the AMM equations of motion [Eq. (6)],

$$\dot{x} = f(x,\alpha,t) \quad ;$$

where t_o (the initial time), t_f (the final time), and $x(t_o)$ are given, and $\Omega(x,t)$ represents the set of admissible controls.

The necessary conditions for an extremal solution to this problem have been given by Breakwell in Ref. 2. Before proceeding further, several comments about this problem formulation are appropriate.

(1) For the computer program that has been developed, $|\alpha(t)|$ is constrained to be less than or equal to 45° or that angle of attack which yields a lateral acceleration maneuver of 50 g [this angle, which will be denoted by $\bar{\alpha}$, is a function of λ and can be calculated from Eq. (4)]. Thus,

$$|\alpha(t)| \leq \min (45^\circ, \bar{\alpha}) \quad . \tag{9}$$

The above relationship defines the set of admissible controls $\Omega(x,t)$ as denoted in (7). Of course, the computer program can incorporate constraints on the angle of attack other than those given in (9).

(2) Expressing the performance criterion J in terms of the final state $x(t_f)$, as in (8), entails no loss of generality. For example, if an integral

$$\int_{t_o}^{t_f} f_o(x,\alpha,t)dt$$

is to be minimized, introduce an additional state x_o satisfying the differential equation

$$\dot{x}_o = f_o(x,\alpha,t) \quad .$$

Note that $f_o(x,\alpha,t)$ is the integrand of the above integral. Then $x_o(t_f)$ is the quantity to be minimized with $x_o(t_o) = 0$.

(3) Terminal constraints on the AMM state have not been considered explicitly in this problem formulation. However, terminal constraints can be treated as follows. If the p-dimensional (where $p < 4$) vector function

$$\psi[x(t_f)] = 0 \tag{10}$$

represents the terminal constraints, then define a new performance criterion

$$J'[x(t_f)] = J[x(t_f)] + K||\psi[x(t_f)]||^2 \ , \quad K > 0 \tag{11}$$

where $J[x(t_f)]$ is the actual performance criterion to be minimized. Thus, a minimization problem subject to terminal constraints is converted to another minimization problem without constraints. With K chosen to be suitably large, it is reasonable to expect that the minimization of $J'[x(t_f)]$ will cause the terminal constraint in Eq. (10) to be satisfied

to an arbitrary degree of closeness. This approach to handling terminal constraints is generally referred to as the "method of penalty functions," and is described in more detail in Ref. 3.

(4) It is not necessary for the AMM initial state $x(t_o)$ to be completely specified. In this case the unspecified initial states can be determined together with $\alpha(t)$ to minimize the performance criterion. For example in the launch doctrine problem, it is important to choose the optimal direction of the initial velocity vector, where the magnitude of the initial velocity is fixed.

(5) In this problem formulation it has been assumed that the final time t_f is fixed. This assumption was made in order to simplify the optimization procedure that has been programmed and is described in Sec. D. Another computer program is being developed which will solve optimization problems with "free" final time.

For the results that will be presented in Sec. E, the performance criterion is a combination of miss distance of the AMM from the target and crossing angle of the AMM relative to the target trajectory:

$$J[x(t_f)] = \underbrace{[y(t_f) - a]^2 + [z(t_f) - b]^2}_{d^2(t_f)} + w \tan^2 \gamma(t_f), \quad w \geq 0 \quad , \tag{12}$$

where

a, b = Position coordinates of the target (ft)

d = Miss distance of the AMM from the target (ft)

γ = $\gamma_T - \gamma_M$ = Crossing angle of the AMM relative to the target trajectory

γ_T = Angle of the target trajectory (above the horizontal axis)

γ_M = Angle of the AMM trajectory (above the horizontal axis).

If the angle γ_T is given by

$$\gamma_T = \tan^{-1}\left(\frac{v_b}{v_a}\right) \quad ,$$

it can be shown that

$$\tan \gamma = \frac{\dot{y}v_b - \dot{z}v_a}{\dot{y}v_a - \dot{z}v_b} \quad . \tag{13}$$

D. Gradient Method

The optimization problem formulated in Sec. C can be solved by using a gradient method that is generally referred to as a steepest-descent technique. This computational procedure, which is described below, is very similar to a technique described by Bryson and Denham in Ref. 4. The steps involved in the gradient method, which has been programmed, are the following.

Step (1) With a control history $\alpha^i(t)$, integrate the AMM equations of motion (6) from the initial state $x(t_o)$. This yields a trajectory $x^i(t)$ and the cost $J^i = J[x^i(t_f)]$.

Step (2) Integrate the adjoint equations

$$\dot{\lambda}^i = -[F^i(t)]^T \lambda^i \tag{14}$$

with boundary conditions

$$\lambda^i(t_f) = \left.\frac{\partial J}{\partial x}\right|_{x^i(t_f)} , \tag{15}$$

where λ^i is the four-dimensional vector containing the adjoint variables. It should be noted that the adjoint equations (14) are integrated backward in time from the boundary conditions given in Eq. (15). The 4 × 4 matrix $F^i(t)$ is obtained by linearizing the AMM equations of motion (6) about the trajectory obtained in Step (1); *i.e.*,

$$F^i(t) = \left.\frac{\partial f}{\partial x}\right|_{x^i(t), \alpha^i(t), t} . \tag{16}$$

From Eqs. (5) and (6), it can be shown that this matrix is given by

$$F^t(t) = \left.\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & f_{32} & f_{33} & f_{34} \\ 0 & f_{42} & f_{43} & f_{44} \end{bmatrix}\right|_{x^t(t),\alpha^t(t),t} \tag{17}$$

where

$$f_{32} = \frac{S}{m}\left(-\frac{1}{2}\frac{\partial\rho}{\partial z}C_D V\dot{y} - \frac{\rho}{2}\frac{\partial C_D}{\partial M}\frac{\partial M}{\partial z}V\dot{y} - \frac{1}{2}\frac{\partial\rho}{\partial z}C_L V\dot{z} - \frac{\partial C_L}{\partial M}\frac{\partial M}{\partial z}V\dot{z}\right)$$

$$f_{33} = \frac{1}{m}\left[S\frac{\rho}{2}\left(-C_D\frac{V^2+\dot{y}^2}{V} - \frac{\partial C_D}{\partial M}\frac{M}{V}\dot{y}^2 - C_L\frac{\dot{y}\dot{z}}{V} - \frac{\partial C_L}{\partial M}\frac{M}{V}\dot{y}\dot{z}\right)\right.$$

$$\left. + T\left(\frac{\dot{z}^2}{V^3}\cos\alpha + \frac{\dot{y}\dot{z}}{V^3}\sin\alpha\right)\right]$$

$$f_{34} = \frac{1}{m}\left[S\frac{\rho}{2}\left(-C_D\frac{\dot{y}\dot{z}}{V} - \frac{\partial C_D}{\partial M}\frac{M}{V}\dot{y}\dot{z} - C_L\frac{V^2+\dot{z}^2}{V} - \frac{\partial C_L}{\partial M}\frac{M}{V}\dot{z}^2\right)\right.$$

$$\left. - T\left(\frac{\dot{y}\dot{z}}{V^3}\cos\alpha + \frac{\dot{y}^2}{V^3}\sin\alpha\right)\right]$$

$$f_{42} = \frac{S}{m}\left(-\frac{1}{2}\frac{\partial\rho}{\partial z}C_D V\dot{z} - \frac{\rho}{2}\frac{\partial C_D}{\partial M}\frac{\partial M}{\partial z}V\dot{z} + \frac{1}{2}\frac{\partial\rho}{\partial z}C_L V\dot{y} + \frac{\rho}{2}\frac{\partial C_L}{\partial M}\frac{\partial M}{\partial z}V\dot{y}\right)$$

$$f_{43} = \frac{1}{m}\left[S\frac{\rho}{2}\left(-C_D\frac{\dot{y}\dot{z}}{V} - \frac{\partial C_D}{\partial M}\frac{M}{V}\dot{y}\dot{z} + C_L\frac{V^2+\dot{y}^2}{V} + \frac{\partial C_L}{\partial M}\frac{M}{V}\dot{y}^2\right)\right.$$

$$\left. + T\left(-\frac{\dot{y}\dot{z}}{V^3}\cos\alpha + \frac{\dot{z}^2}{V^3}\sin\alpha\right)\right]$$

$$f_{44} = \frac{1}{m}\left[S\frac{\rho}{2}\left(-C_D\frac{V^2+\dot{z}^2}{V} - \frac{\partial C_D}{\partial M}\frac{M}{V}\dot{z}^2 + C_L\frac{\dot{y}\dot{z}}{V^3} + \frac{\partial C_L}{\partial M}\frac{M}{V}\dot{y}\dot{z}\right)\right.$$

$$\left. + T\left(\frac{\dot{y}^2}{V^3}\cos\alpha - \frac{\dot{y}\dot{z}}{V^3}\sin\alpha\right)\right] .$$

Using Eqs. (12) and (13), it can be shown that the boundary conditions in Eq. (15) are given by

$$\lambda'(t_f) = \begin{bmatrix} 2(y - a) \\ 2(z - b) \\ w\,\dfrac{2\eta(\dot{y}v_b^2 - \dot{z}v_a v_b) - 2\nu(\dot{y}v_a^2 + \dot{z}v_a v_b)}{\eta^2} \\ w\,\dfrac{2\eta(-\dot{y}v_a v_b + \dot{z}v_a^2) - 2\nu(\dot{y}v_a v_b + \dot{z}v_b^2)}{\eta^2} \end{bmatrix}_{x'(t_f)} \tag{18}$$

where

$$\nu = (\dot{y}v_b - \dot{z}v_a)^2$$

$$\eta = (\dot{y}v_a + \dot{z}v_b)^2 \quad .$$

Step (3) Having th adjoint variables $\lambda'(t)$ obtained in Step (2), the gradient of the performance criterion with respect to the control can be calculated as

$$\nabla_\alpha J'(t) = [G'(t)]^T \lambda'(t) \quad , \tag{19}$$

where the 4×1 matrix $G'(t)$ is obtained by linearizing the AMM equations of motion about the trajectory obtained in Step (1); *i.e.*,

$$G'(t) = \left.\frac{\partial f}{\partial \alpha}\right|_{x'(t),\alpha'(t),t} \quad . \tag{20}$$

From Eq. (6) it can be shown that this matrix is given by

$$G'(t) = \begin{bmatrix} 0 \\ 0 \\ g_3 \\ g_4 \end{bmatrix}_{x'(t),\alpha'(t),t} \tag{21}$$

where

$$g_3 = \frac{1}{m}\left[S\frac{\rho}{2}\left(-\frac{\partial C_D}{\partial \alpha}V\dot{y} - \frac{\partial C_L}{\partial \alpha}V\dot{z}\right) - T\left(\frac{\dot{y}}{V}\sin\alpha + \frac{\dot{z}}{V}\cos\alpha\right)\right]$$

$$g_4 = \frac{1}{m}\left[S\frac{\rho}{2}\left(-\frac{\partial C_D}{\partial \alpha}V\dot{z} + \frac{\partial C_L}{\partial \alpha}V\dot{y}\right) + T\left(-\frac{\dot{z}}{V}\sin\alpha + \frac{\dot{y}}{V}\cos\alpha\right)\right] \quad .$$

Step (4) Using the gradient obtained in Step (3), compute a new control history

$$\alpha^{i+1}(t) = \alpha^{i}(t) - \mu^{i}[G^{i}(t)]^{T}\lambda^{i}(t), \qquad \mu^{i} > 0 \quad . \tag{22}$$

The step-size μ^i is determined so that $J^{i+1} < J^i$, where the cost J^{i+1} results from the control $\alpha^{i+1}(t)$. The quantity μ^i is also chosen to increase the convergence rate of the gradient method.

Steps (1) through (4) are repeated until this iterative procedure converges; *i.e.*, until the gradient in Eq. (19) tends to zero. Hence, the gradient method can only be guaranteed to yield a relative minimum for the cost J. To start the gradient method, an initial control history $\alpha^0(t)$ must be provided. In general, a reasonable control history $\alpha^0(t)$ can be chosen from physical considerations. Indeed, the proper choice of $\alpha^0(t)$ can ensure that this iterative procedure will converge to the absolute minimum of J. Finally, it should be noted that the gradient method described above yields an optimum control history, which is "open loop" (*i.e.*, the angle-of-attack commands are not based on the AMM state), and the corresponding optimal trajectory for the AMM.

E. Results

The gradient method, which is described in Sec. D, has been programmed in ALGOL and run on a B-5500 computer. To solve the AMM equations of motion (6) and the adjoint equations (14), an Adams-Moulton (4-point) integration routine, with a constant time increment of 0.1 second, has been employed. The program has been used to compute optimal guidance histories (and the resulting optimal trajectories) for the performance criterion given in Eq. (12). For this problem, the program requires approximately four seconds per iteration.

Several examples have been run using a set of aerodynamic tables for the drag coefficient C_D and the lift coefficient C_L that are representative of a short-range AMM. The five cases described in this report are summarized in Table 1, which gives target position, target-trajectory angle, and

Table 1

TARGET PARAMETERS FOR THE COMPUTER RUNS

	CASE 1	CASE 2	CASE 3	CASE 4	CASE 5
a(ft) b(ft)	45×10^3 50×10^3	70×10^3 40×10^3	70×10^3 40×10^3	70×10^3 40×10^3	70×10^3 40×10^3
γ_T	--	--	90.0°	63.5°	26.5°
w	--	--	10^9	10^9	$10^7, 10^8, 10^9$

weighting factor w for the performance criterion J, as defined in Eq. (12). In Cases 1 and 2, the cost J only contains a term for miss distance; in Cases 3 and 4, the cost J consists of terms for miss distance and crossing angle, with a constant weighting factor; in Case 5, the cost J consists of terms for miss distance and crossing angle, for different weighting factors. For these cases the following system parameters have been used:

$$
\begin{aligned}
x(t_o) &\begin{cases} y(t_o) = 2.193 \times 10^4 \text{ ft} \\ z(t_o) = 8.895 \times 10^3 \text{ ft} \\ \dot{y}(t_o) \quad 8.970 \times 10^3 \text{ ft/sec} \\ \dot{z}(t_o) = 2.290 \times 10^3 \text{ ft/sec} \end{cases} \\
t_f - t_o &= 10.0 \text{ sec} \\
T &= 0 \text{ lb} \\
m &= 32.78 \text{ slugs} \\
S &\quad 1.355 \text{ ft}^2 \quad .
\end{aligned}
\tag{23}
$$

The initial state $x(t_o)$ corresponds to the state of a short-range AMM immediately after engine burnout; after t_o, thrust is equal to zero and mass is equal to a constant.

The results generated by the gradient-method program for these cases are presented in Figs. 2 through 6. In Figs. 2(a), 3(a), 4(a), and 5(a), the control histories for intermediate iterations are shown; between successive

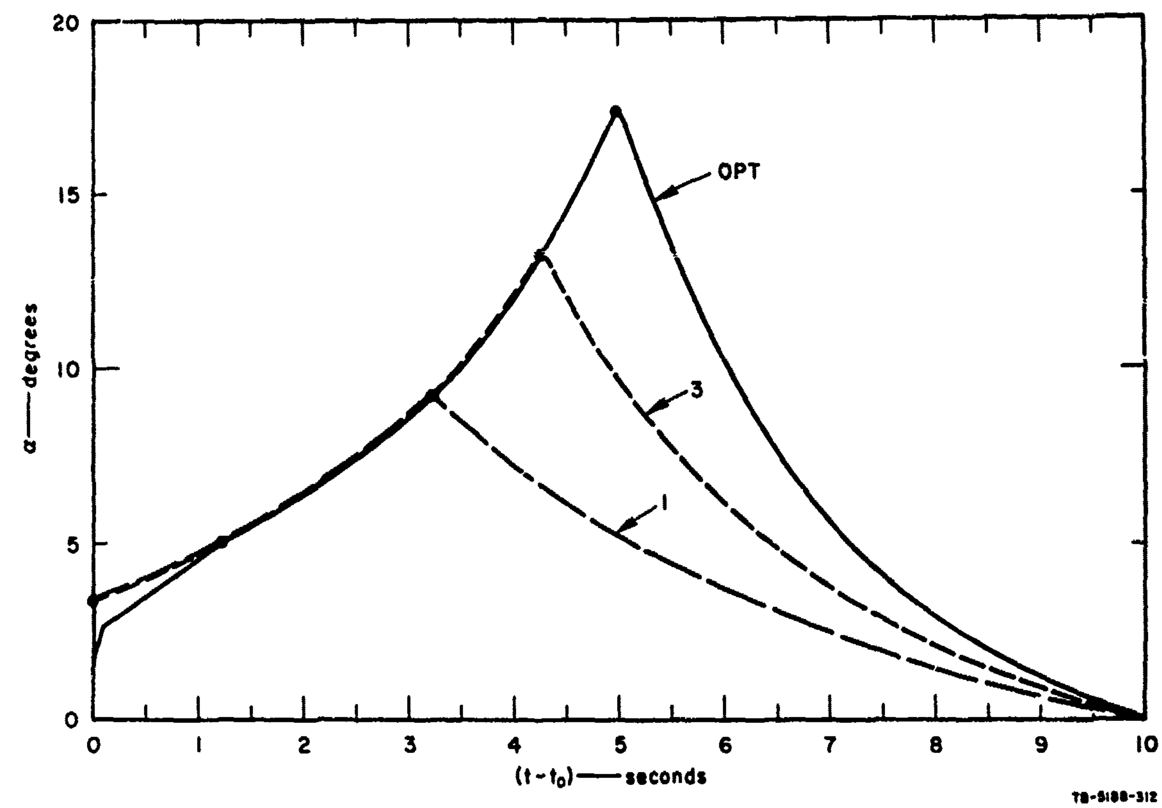

FIG. 2(a) ANGLE-OF-ATTACK HISTORIES FOR CASE 1

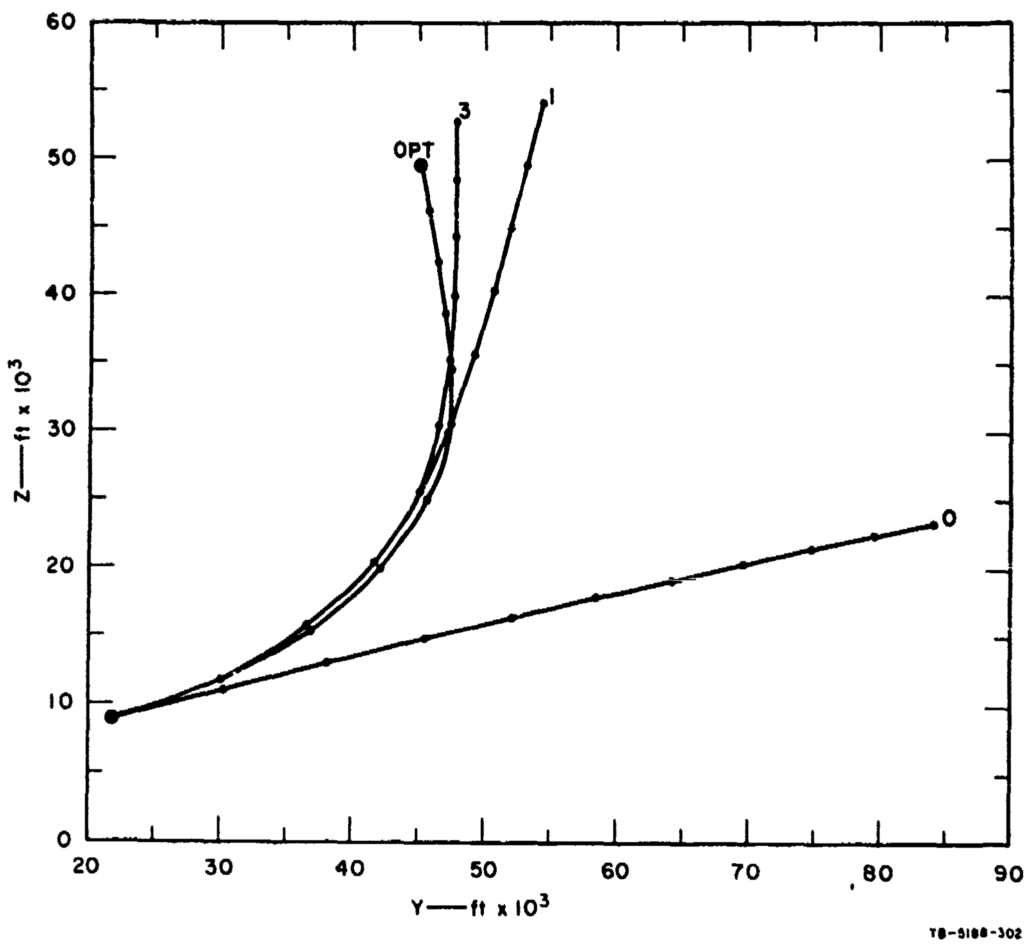

FIG. 2(b) TRAJECTORIES IN THE y-z PLANE FOR CASE 1

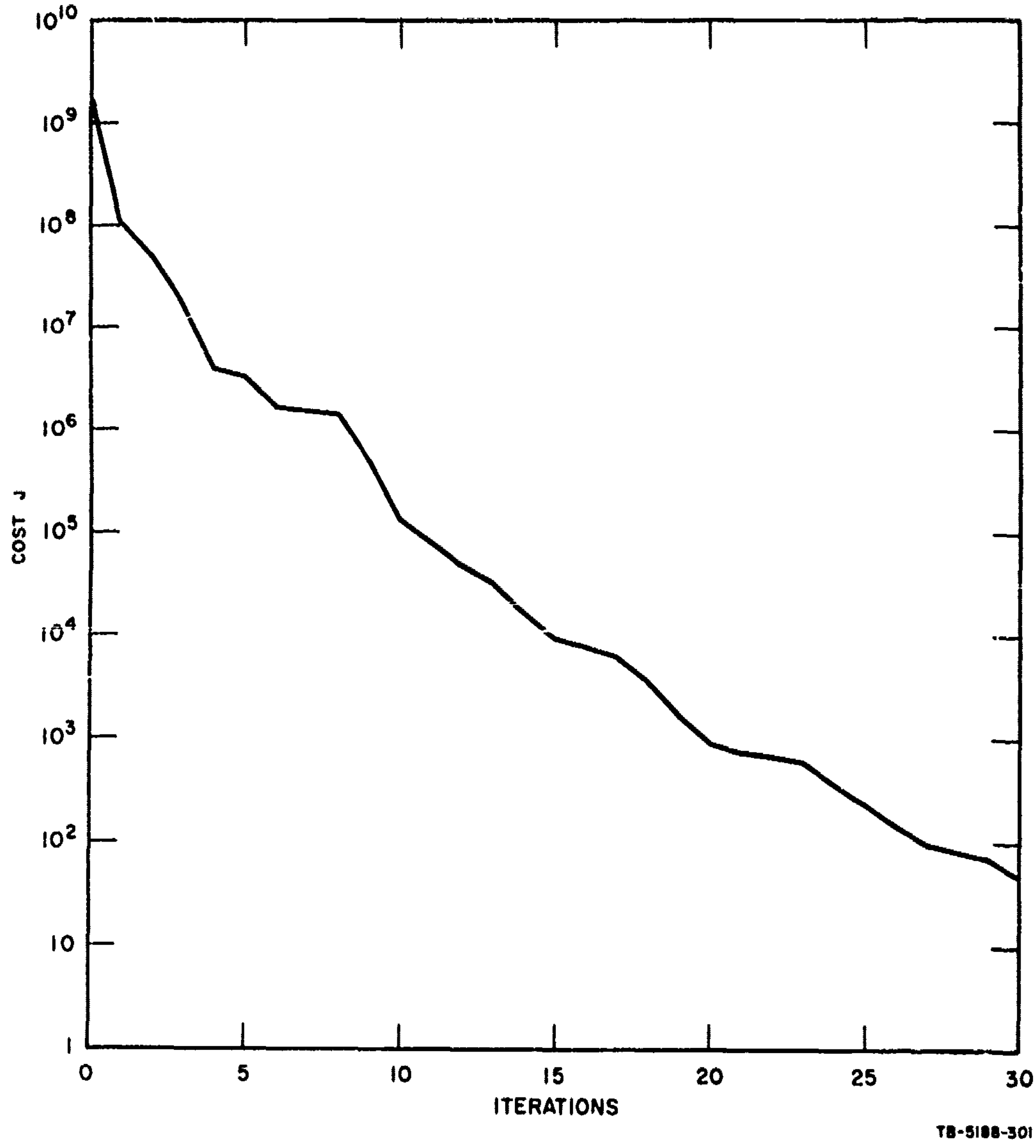

FIG. 2(c) COST vs. ITERATIONS FOR CASE 1

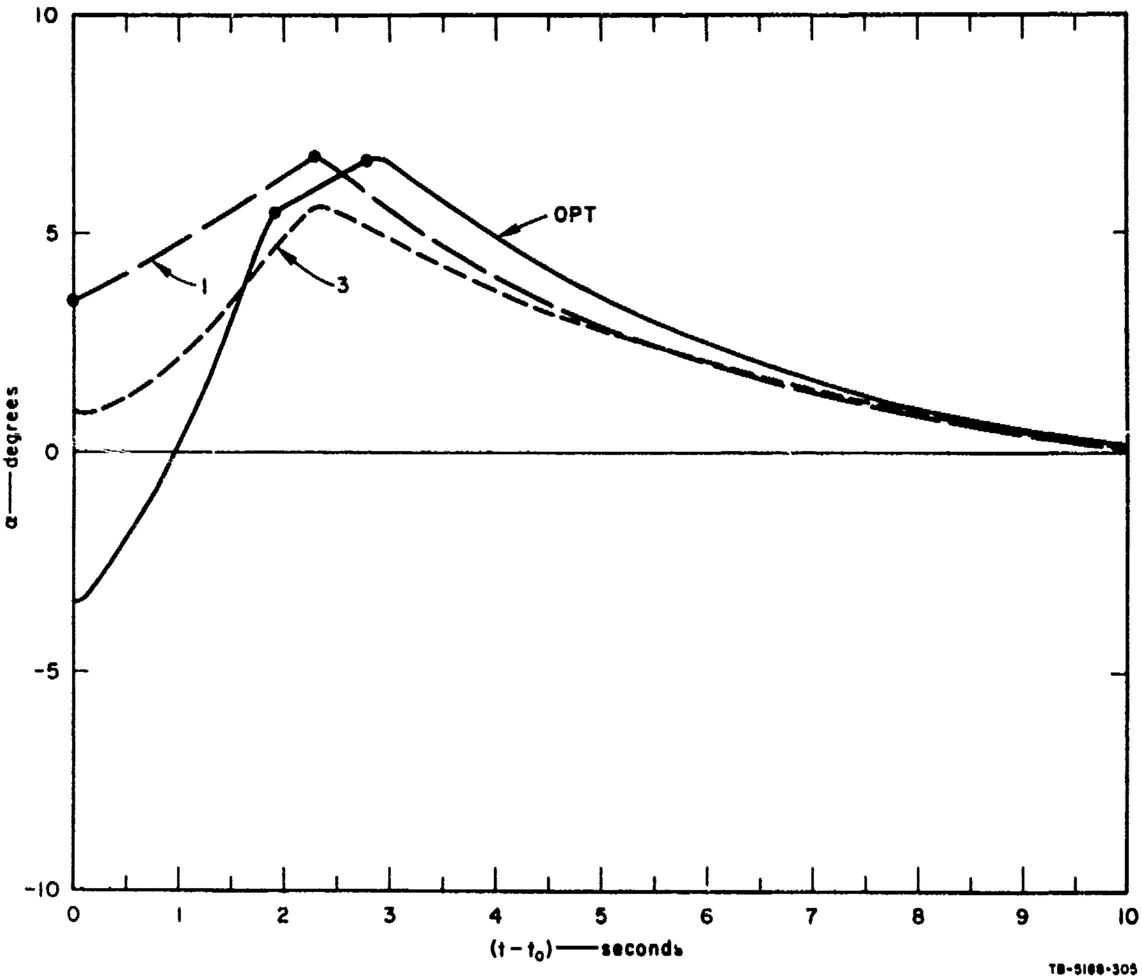

FIG. 3(a) ANGLE-OF-ATTACK HISTORIES FOR CASE 2

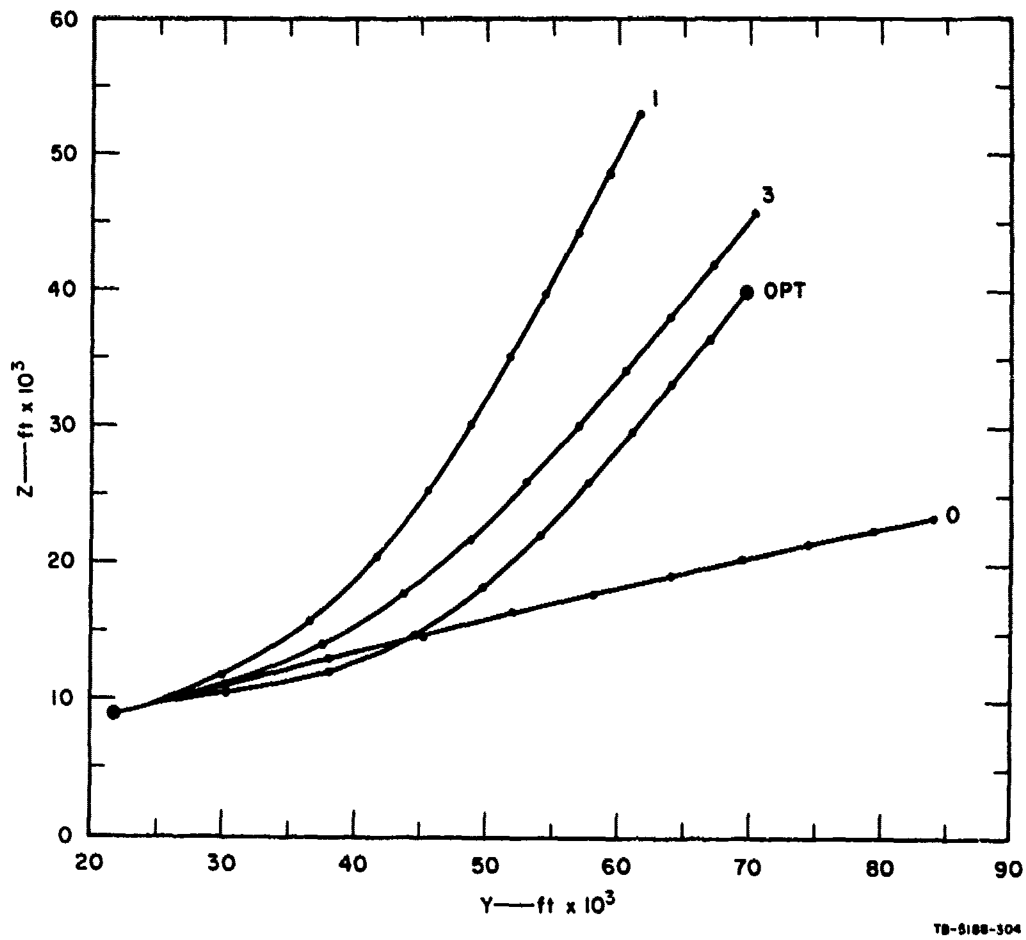

FIG. 3(b) TRAJECTORIES IN THE y–z PLANE FOR CASE 2

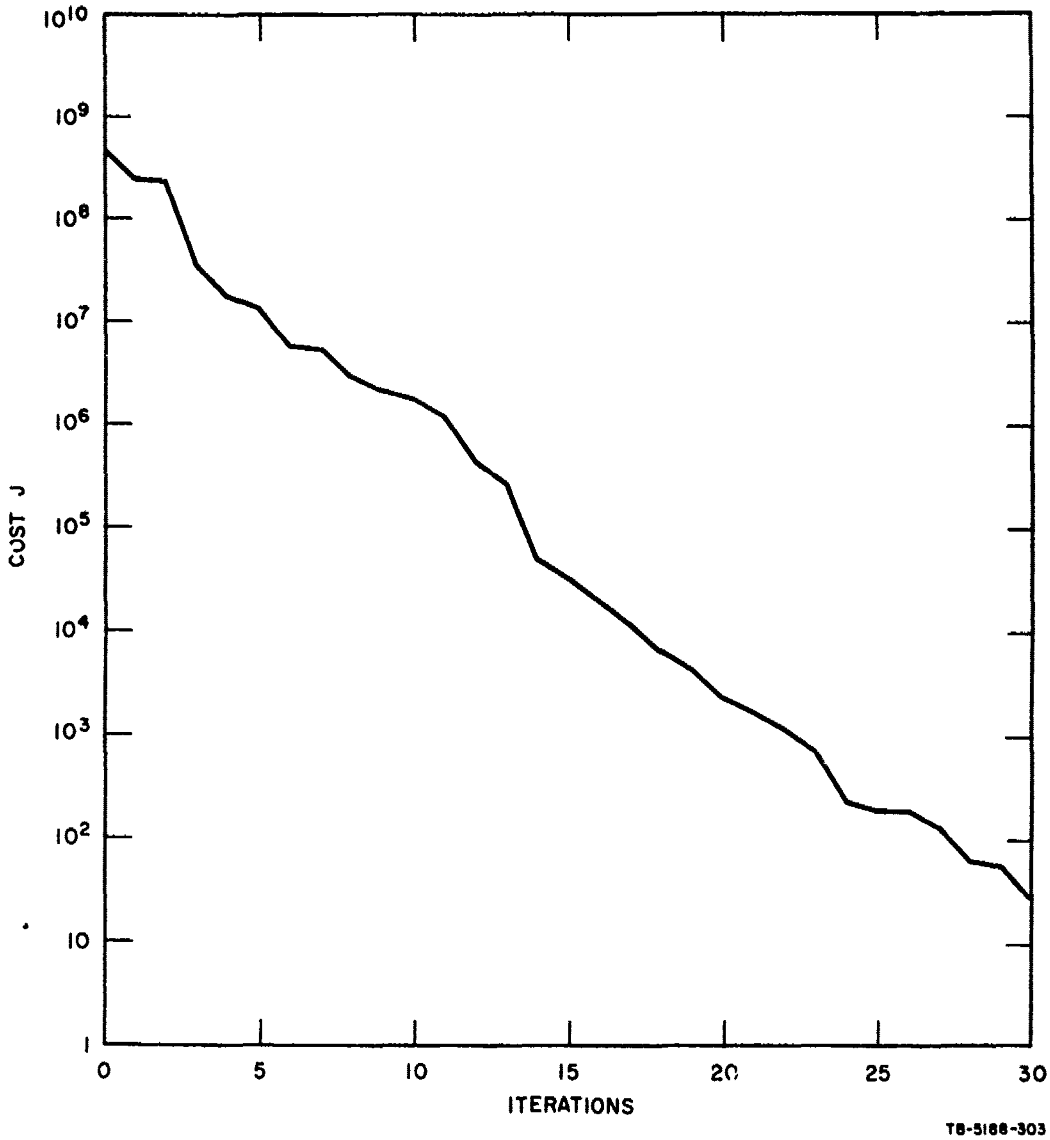

FIG. 3(c) COST vs. ITERATIONS FOR CASE 2

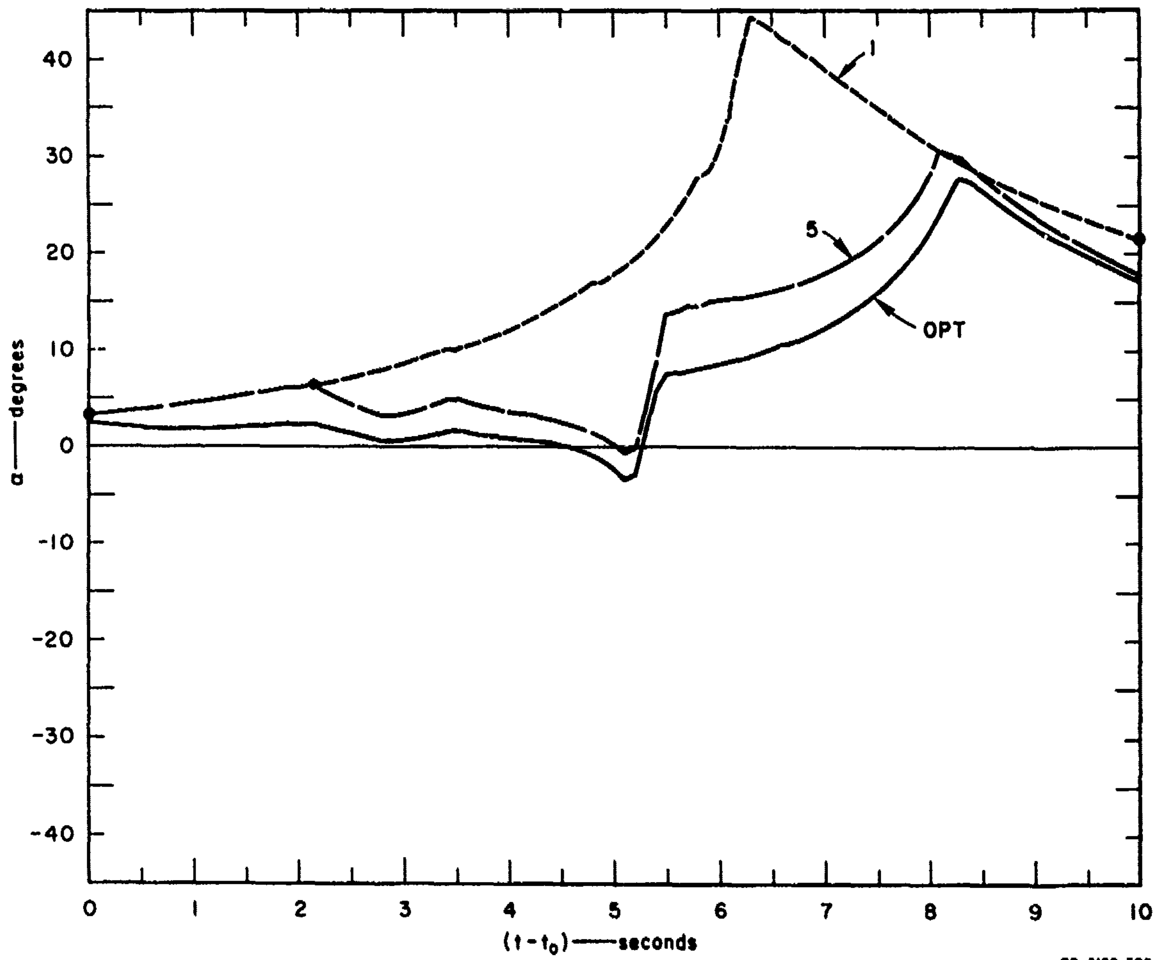

FIG. 4(a) ANGLE-OF-ATTACK HISTORIES FOR CASE 3

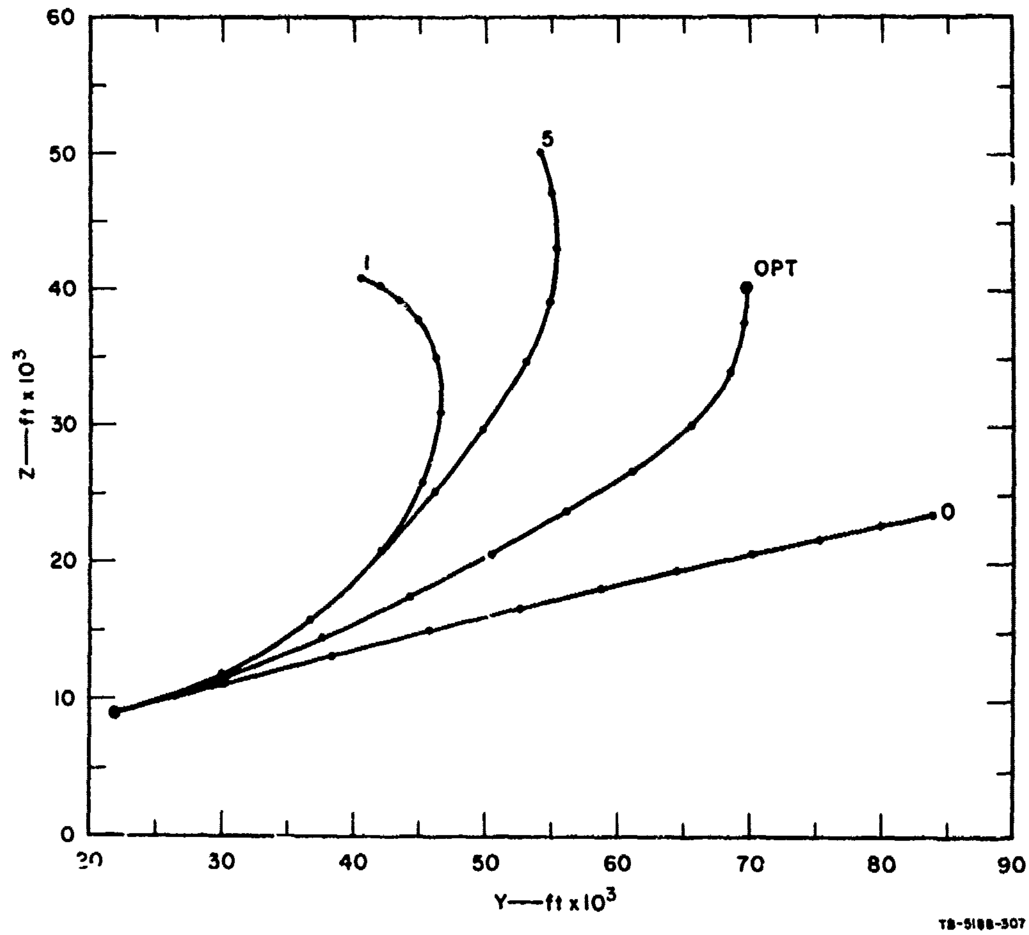

FIG. 4(b) TRAJECTORIES IN THE y–z PLANE FOR CASE 3

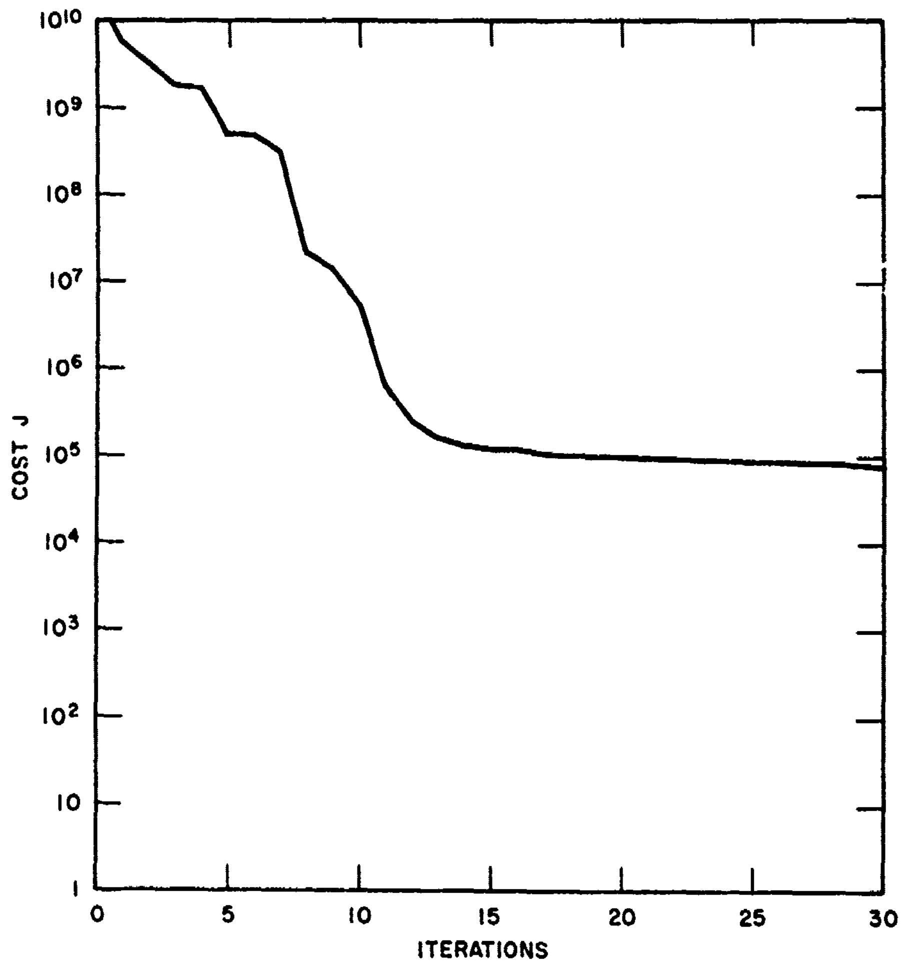

FIG. 4(c) COST vs. ITERATIONS FOR CASE 3

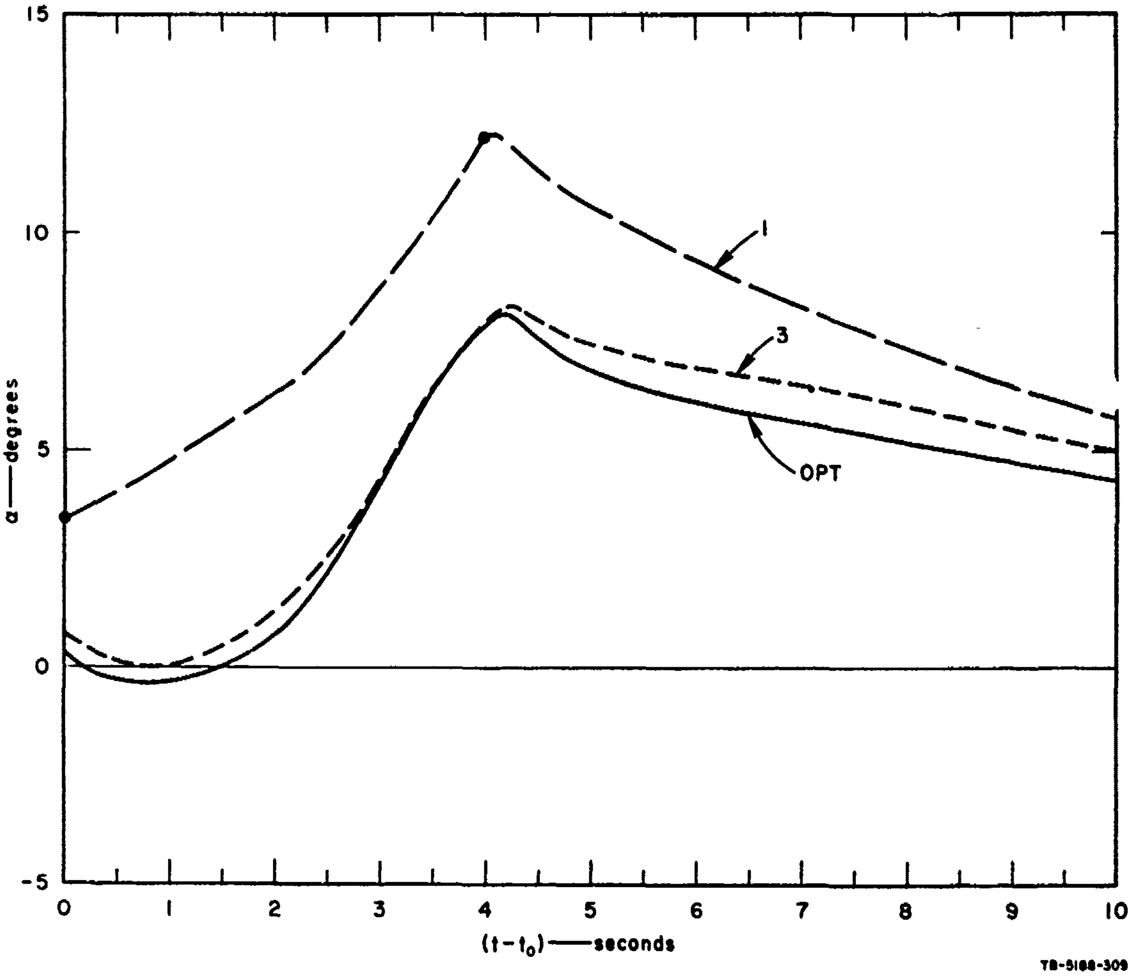

FIG. 5(a) ANGLE-OF-ATTACK HISTORIES FOR CASE 4

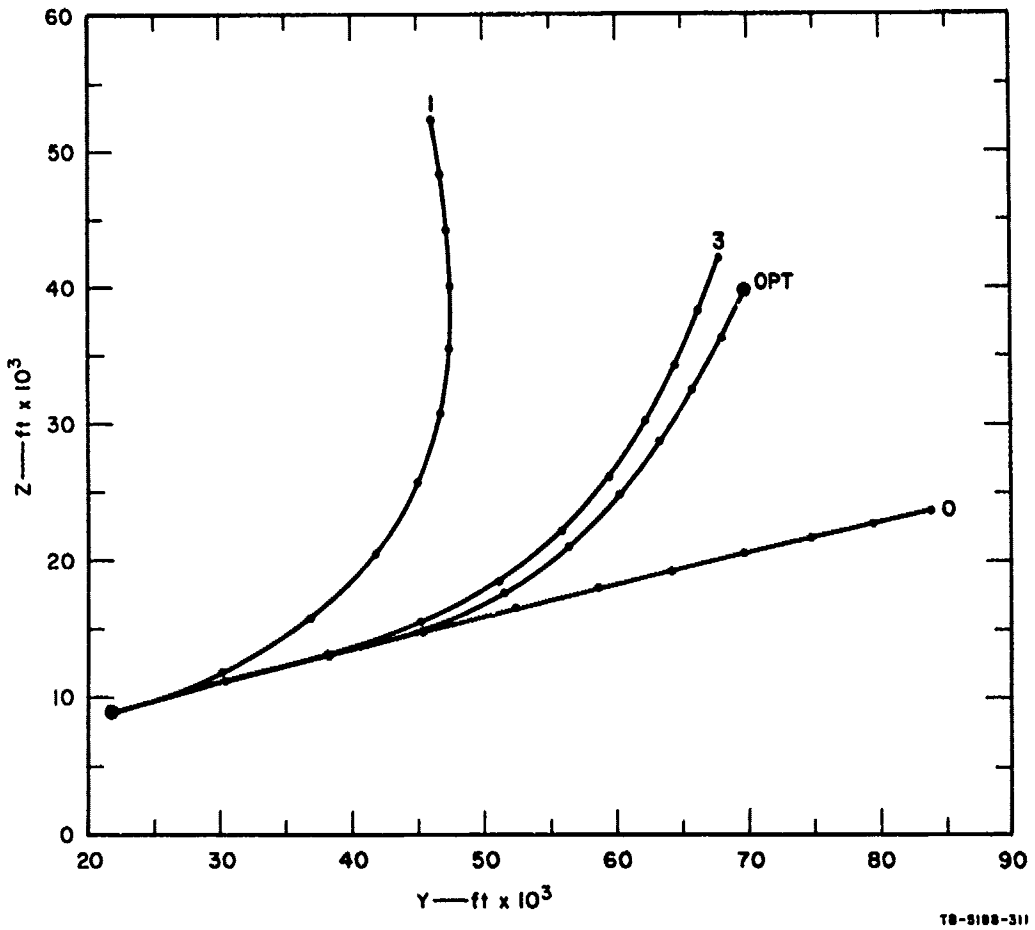

FIG. 5(b) TRAJECTORIES IN THE y-z PLANE FOR CASE 4

FIG. 5(c) COST vs. ITERATIONS FOR CASE 4

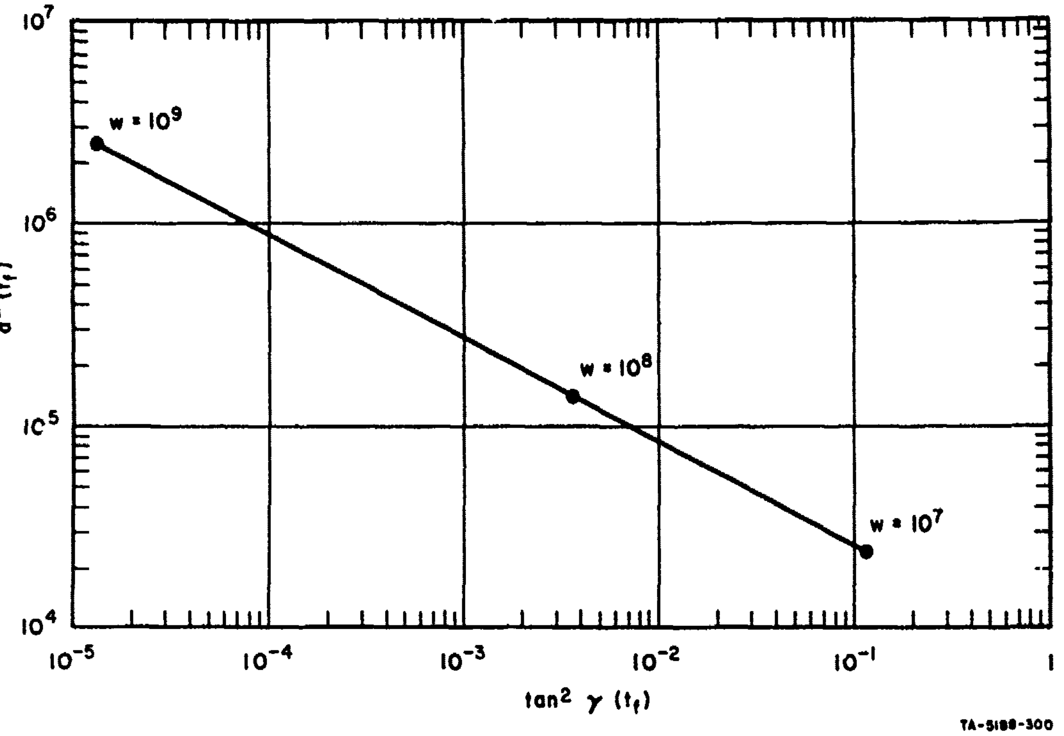

FIG. 6 RESULTS FOR CASE 5: MISS DISTANCE vs. CROSSING ANGLE

dots the angle of attack is limited by the constraint given in (9). In Figs. 2(b), 3(b), 4(b), and 5(b), the trajectories in the y - z plane for intermediate iterations are given; the dots represent one-second intervals in time. The curves labeled OPT in these figures correspond to the solutions to which the gradient method has converged. In Figs 2(c), 3(c), 4(c), and 5(c), the performance criterion J is plotted versus the number of iterations. For each value of w in Case 5, the optimal values of $d^2(t_f)$ and $\tan^2 \gamma(t_f)$ were computed using the gradient method. The results which are plotted in Fig. 6, demonstrate the trade-off between miss distance and crossing angle that is obtained by varying the weighting factor w. The slope of the resulting curve in Fig. 6 can be interpreted as being a measure of this trade-off. Of course, this slope will depend on the geometry of the specific problem, *i.e.*, on the initial AMM state and the target parameters.

If the appropriate identifications are made, Cases 3 and 4 can be considered as applications of the method of penalty functions (this approach was described in Sec. C) to problems where the terminal constraint to be satisfied is

$$\gamma(t_f) = 0 \quad , \tag{24}$$

in which γ is the crossing angle of the AMM relative to the target trajectory, and the actual performance criterion to be minimized is

$$J = [y(t_f) - a]^2 + [z(t_f) - b]^2 = d^2(t_f) \quad , \tag{25}$$

in which d is the miss distance of the AMM from the target. For $w = 10^9$, the gradient method converges to a solution for which the terminal constraint in Eq. (24) is nearly satisfied and the cost in Eq. (25) is minimized. The results for Case 3 are

$$\begin{aligned} \gamma(t_f) &= 1.7 \times 10^{-3} \text{ rad} \\ d(t_f) &= 122 \text{ ft}; \end{aligned} \tag{26}$$

and for Case 4

$$\begin{aligned} \gamma(t_f) &= 0.4 \times 10^{-3} \text{ rad} \\ d(t_f) &= 16 \text{ ft} \quad . \end{aligned} \tag{27}$$

For all of these computer runs, the gradient method was initialized with $\alpha'(t) \quad 0$. As can be seen from Figs. 2 through 5, this choice for the initial control history (and the resulting initial trajectory) differs markedly from the optimal solutions. As these figures indicate, the gradient method initially has a rapid convergence rate [see, in particular, Figs. 2(c), 3(c), 4 (c), and 5(c)]. However, as shown in Figs. 4(c) and 5(c), the convergence rate of the gradient method is relatively slow when the optimum is being approached.

F. Conclusions and Summary

In this study a numerical procedure for computing optimal guidance commands for an AMM to optimize a given performance criterion has been described. The problem formulation is presented in Sec. C, and the gradient method that has been programmed is detailed in Sec. D.

The usefulness of this computational technique is demonstrated in Sec. E, where numerical results for several examples are described. The gradient-method program is a valuable tool for future studies in the area of AMM guidance and control. It can be used for the purpose of evaluating suboptimal schemes that are proposed; *i.e.*, it can provide a "yardstick" for performance comparisons. In addition, by modifying the basic gradient method, it may be possible to develop on-line computational procedures for generating control commands for the AMM. This is particularly important in obtaining real-time operation, since only a few iterations may be required.

The gradient method is particularly well suited to optimizing about a given nominal trajectory. Hence, it can be applied to the control problem after an initial trajectory has been obtained at prelaunch.

REFERENCES

1. W. G. Keckler and R. E. Larson, "Dynamic Programming for Prelaunch," Memorandum 25, Contract DA-01-021-AMC-90006(Y), SRI Project 5188-233, Stanford Research Institute, Menlo Park, California (October 1966).

2. J. V. Breakwell, "The Optimization of Trajectories," *Journal of the Society of Industrial and Applied Mathematics*, Vol. 7, No. 2, pp. 215-247 (June 1959).

3. H. J. Kelley, "Method of Gradients," *Optimization Techniques*, G. Leitmann, ed., Chapter 6, pp. 205-254, (Academic Press, New York, 1962).

4. A. E. Bryson and W. F. Denham, "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Transactions of the ASME, Journal of Applied Mechanics*, Vol. 29, No. 3, pp. 247-259 (June 1962).

Chapter IV

APPLICATION OF A GAMING APPROACH TO MRV INTERCEPTION

A Introduction

Successful interception of an attacking missile depends heavily on the performance characteristics of the attacking missile. For a maneuvering reentry vehicle (MRV), the basic characteristics are the following:

(1) The attacking missile is capable of maneuvering while approaching the target.

(2) The attacking missile is *not capable* of observing the interceptor.

(3) The interceptor, on the other hand, is capable of observing the attacker during the reentry phase, using highly sophisticated ground-based radar systems.

Under these assumptions, the question of optimal behavior becomes extremely complicated for both the attacker and interceptor. Roughly speaking, the attacker's problem is in what sense is a maneuver optimal when the interceptor's position is unknown. This problem induces the interceptor's problem—to *predict* how the attacker is going to maneuver and how to maneuver to interception

One possible way to simplify the interceptor's problem is to postulate that the attacker will optimize a given function of his trajectory. Then the optimal attacking trajectory can be determined, and the best countermeasure can be found. This approach, which is essentially equivalent to assuming a nonmaneuvering attacker, is highly weighted in favor of the interceptor. Unfortunately, the resulting interception policy depends strongly on the postulated attacker objective function and may turn out to be extremely bad if the attacker is actually maneuvering according to some other criterion. An intelligent attacker, thus, will not use the postulated objective function. For this reason this approach is not considered further

Another possible way to simplify the interceptor's problem is to relax the second characteristic, and assume that the attacker *is capable* of observing the interceptor. This approach leads to a classical differential game problem, which can be solved by existing techniques. However, the solution is meaningless for the attacker, because he cannot implement it. For the interceptor, however, the solution is a valid pessimistic answer in the sense that it provides a feasible interception control policy that guarantees the interceptor a minimum performance level independent of the attacker's behavior. By ignoring his blindness this approach favors the attacker, and as a result the guaranteed minimum performance may be overly pessimistic. Since attempts in this direction have been discouraging, a major question is how much better can the interceptor perform by taking into account the blindness of the attacker.

The following is an attempt to formulate the problem of MRV interception with the basic characteristics above and to study the result. Several simplified examples are solved, both under this formulation and as a discretized differential game, and the solutions are discussed. The problem of generalizing the method of solution used for these examples is under study, as are several alternative approaches.

B Problem Formulation

Consider one defended area, one attacking MRV, and a defense which may have one or more AMM's for interception. The dynamic characteristics and the number of AMM's are known to both. The attacker must hit the defended point; if he is intercepted successfully, he suffers a positive loss that is a known function of the interception point. The attacker knows the initial positions of the AMM's and his own position at all times, whereas the interceptor knows not only the AMM's positions at all times, but also the attacker's position, if the attacker is within a specified region near the defended point. This region is also known to the attacker. There is assumed to be no noise or delay in observations or calculations and no reliability problem.

The attacker's objective is to minimize his loss, and the interceptor's objective is to maximize the same loss; however, the choice of which side is considered to suffer the loss is arbitrary. As stated above, the attacker's loss is, for example, a measure of how short he is of completely destroying the defended point. If, instead, the direct damage to the defended point were considered as the interceptor's loss,

the only effect would be a switch of minimizer with maximizer. When the attacker is chosen to be the minimizer, and the damage to the defended point is a two-valued function, the expected value of the objective becomes the probability of interception—a convenient measure when the effectiveness of an interceptor is the subject of interest. The assumption implicit in equating probability of noninterception with defended point damage is that the attacker will destroy the defended point if not intercepted. This assumption is valid if the attacker is constrained to use trajectories terminating in the defended point

Now, assume the region observable to the interceptor to be such that once the attacker gets into it, he cannot get out without missing the defended point. Since the interceptor cannot act before the attacker enters this region, this is the only portion of the attacker trajectory that is of interest. Once the attacker chooses a point of entry to this region his problem is to minimize and the interceptor's problem is to maximize, from this point on. Once this problem is solved separately for any boundary point, the attacker is free to choose a best entry point.

To describe the problem mathematically, the following objects must be defined.

(1) Let X_a and X_i be the state spaces of attacker and interceptor, respectively (X_i is the direct sum of the state space of the AMM's if more than one are available). Let $x_a(0)$ and $x_i(0)$ be the respective initial states, and TR be the subset of X_a describing the defended area.

(2) Let $\{R_a\}$ be the set of attacker trajectories R_a starting at $x_a(0)$ and terminating in TR, and let R_a^t be the portion of the trajectory R_a executed up to time t. Similarly, let $\{R_i\}$ be the set of interceptor trajectories starting at $x_i(0)$, and R_i^t be the portion of the trajectory R_i executed up to time t. Note that there is no terminating restriction and that the spaces are different.

(3) Let $\{u_i(x_i, t)\}$ be the set of controls available to the interceptor at time t if it is in state x_i, and let $\{U_i(\ ,\ ,\ )\}$ be a set of functions

$$U_i(R_i^t, R_a^t, t) : \{R_i^t\} \times \{R_a^t\} \to \{u_i[x_i(R_i^t), t]\} \quad ,$$

where $x(R_i^t)$ denotes the interceptor state at time t on the trajectory R_i^t

(4) Finally, let $L(x_a)$ be the loss to the attacker if intercepted at state x_a.

If attacker and interceptors execute trajectories R_a and R_i, respectively, then the attacker state at (first) interception may be calculated (if not intercepted then it may be taken to be on the defended point). Denote this state by $x(R_a, R_i)$. If the attacker executes R_a, and the interceptor uses a function $U_i(.,...)$ as his control law, then a trajectory $R_i(R_a, U_i)$ is defined by the interceptor's dynamics. The problem can be stated now as follows:

> The attacker and interceptor choose $R_a \in \{R_a\}$ and $U_i(.,..,.) \in \{U_i(.,...)\}$ in order to minimize and maximize, respectively, the function
> $$L\{x[R_a, R_i(R_a, U_i)]\} \triangleq L(R_a, U_i) \quad .$$

In this form the MRV interception problem is a one-step, two-player, zero-sum game, where the minimizer's strategies are $\{R_a\}$, the maximizer's strategies are $\{U_i(.,..,.)\}$, and the outcome function is $L(R_a, U_i)$. A pair of strategies, $R_a^*, U_i^*(.,\ ,\ )$, is considered to be a solution if

$$L(R_a^*, U_i) \leq L(R_a^*, U_i^*) \leq L(R_a, U_i^*)$$

$$U_i \in \{U_i(\ .\ ,.)\} \qquad R_a \in \{R_a\}$$

Hence, by choosing the corresponding component of a solution, either player, separately, guarantees that he achieves $L(R_a^*, U_i^*)$, at least, and if any player makes a choice that is not a component of a solution, then his opponent is able to make a choice that improves the outcome for him. It is well known that a necessary and sufficient condition for the existence of a solution is that

$$\min_{R_a \in \{R_a\}} \max_{U_i \in \{U_i(.,...,.)\}} L(R_a, U_i) = \max_{U_i \in \{U_i(.,...,.)\}} \min_{R_a \in \{R_a\}} L(R_a, U_i) \quad ,$$

and any root of this equation is a solution.

It should be noted at this point that the attacker trajectories, R_a, may be described by the controls producing them, which may be considered as functions of time and the portion of the trajectory already completed, but not of the interceptor's trajectory. Thus, the set $\{R_a\}$ may be equivalently replaced by a suitable set of control laws:

$$\{U_a(\cdot,\cdot)\} \cdot U_a(R_a^t, t) : \{R_a^t\} \to u_a[x_a(R_a^t), t] \quad ,$$

thus making the problem appear slightly more symmetrical.

The problem above was put in terms of trajectories and portions of trajectories, in order to provide a formulation for what is intuitively the problem of the ultimate best each side can do. The attacker is bound to choose his trajectory in advance, and all he can do is to optimize on this choice, and the ultimate best the interceptor can do is to use, wisely, all the information about past trajectories. However, for practical reasons, it may be appropriate to restrict the players further by requiring that the applied controls depend only on the present states rather than the entire past trajectories. Whether or not this restriction changes the game is an important question, and while it seems trivial that it makes no difference if this constraint is imposed on the attacker, it is not in general true that such a constraint makes no difference for the interceptor. With such restrictions, a new game may be considered in a single state space, $X_a \oplus X_i$. This new game is a differential game with the state-space *not* completely observable to one of the players. Unfortunately, no general method of solution exists for this kind of game.

C. Existence of Solutions

In general, there is no guarantee that a solution exists to the above game in any of the versions. On the contrary, we would expect intuitively a solution *not* to exist, because existence implies the possibility of calculating the attacker's trajectory in advance. The standard approach in such situations is to enlarge the given sets of strategies by permitting randomized actions. The new strategies, which are called mixed to distinguish them from the old or pure strategies, are probability distributions on the sets of pure strategies. By considering the distributions where probability one is assigned to one pure strategy, pure strategies may be considered as special cases of mixed strategies.

The sets of pure strategies for attacker and interceptor are thus replaced by the larger sets of mixed strategies. A player's choice in the game becomes the selection of one mixed strategy from his set. The outcome of both players' choice is the *expected value* of the probability distribution thus induced on the outcome-function values. In practice the player then will make a selection with a randomness that is defined by the mixed strategy, and execute the pure strategy that results.

Ignoring some delicate mathematical difficulties, which are not of interest here and which can be taken care of, we may assume that an at least approximate mixed solution always exists.

D. A Simplified Case

A drastic simplification of the problem is achieved by quantizing the state spaces. We consider the following case:

> The state spaces are discrete sets of points, and the players simultaneously move in discrete times. For each point there is a given set of points reachable in one move, thus defining the set of available controls. We also assume that all the attacker's permitted trajectories (*i.e.*, those starting at the given initial point and terminating in the target) are uniformly bounded in length.

Under these assumptions, the sets of pure strategies for each player become finite, and there is an optimal solution. In principle, the solution can be found by writing down the resulting game matrix and using well-known techniques. However, the matrix may be found to be impractically large—indeed, even for the extremely simplified examples presented in the next section the matrix is so big that it can not be written down. A technique that has been developed for solving these examples may be described in the following general terms. The attacker's available controls are described on his state space as a directed graph, *i.e.*, a collection of arrows describing permitted moves. Suppose the attacker chooses his mixed strategy and consider a minimal cut set separating the target from the initial condition, *i.e.*, a collection of arrows that when removed from the graph disconnects the graph between initial condition and defended point Let p_i be the probability that the attacker will pass through the ith arrow according to his mixed strategy. Select any state that the interceptor may occupy at that time, say the kth, and find his best action, given that he knows the values p_i. Let the outcome be $l_k(p_1, \ldots p_i, \ldots)$. For the attacker's mixed strategy to be a solution, the p_i must satisfy

$$\min_{\substack{\text{all possible values} \\ \text{of } (p_1, \ldots p_i, \ldots)}} \max_k \; l_k(p_1, \ldots p_i, \ldots) \quad ,$$

and the value obtained is a lower bound to the value of the game, *i.e.*, the outcome of an optimal play. The required minimax can then be obtained by linear programming.

By carefully choosing the appropriate cut sets in appropriate order, and matching the probabilities obtained on the boundary points, the class of "worthwhile" strategies is so drastically reduced that the solution(s) can be found. The matching phase requires finding the set of all solutions of a linear program; a method for doing this has been developed.

E. Examples

For the examples, consider a two-dimensional interception space with a single defended point. The state of any involved missile (attacker or any of the AMMs) is taken to be its position, which is restricted to be at any point of a rectangular, two-dimensional grid. Thus the state spaces of all involved missiles will be described on the same grid. The set of controls available to any missile is the same in any state, and is illustrated by Fig. 1. Further, in all the examples the attacker's initial state is four units above the defended point. Then the set $\{R_a\}$ of all qualified trajectories is found to be as described by the solid graph (Fig. 2). The "broken" arrows represent permissible controls that cannot be used because they cannot be continued to trajectories terminating in the defended point.

The initial position for any AMM is on the defended point, and an interception is considered to occur if both missiles occupy the same position simultaneously or move along the same segment at the same time. Thus, an AMM move is described on the same graph, with reversed arrows. When in initial position, an AMM is permitted an extra "zero" control, *i.e.*, to remain in place, thus enabling choice of the best time to launch. The loss function is taken to be two-valued (0/1), which makes the probability of interception the value of the game. The examples will differ with respect to where the loss is 1, and with respect to the number of AMM's available to the defense.

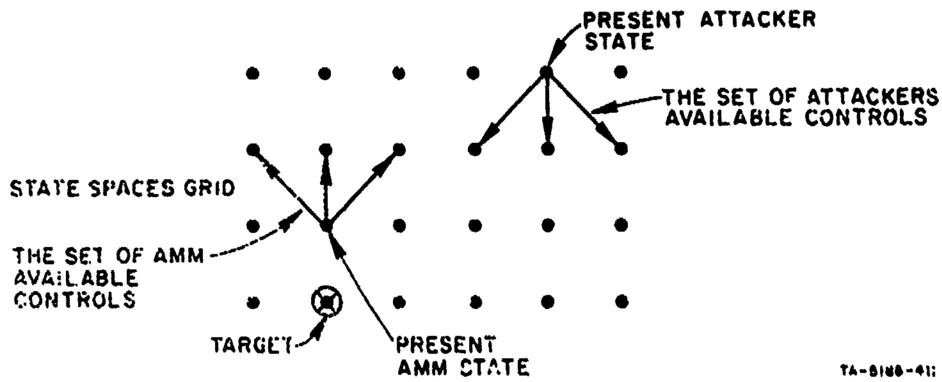

FIG. 1 AVAILABLE CONTROLS FOR AMM AND ATTACKER

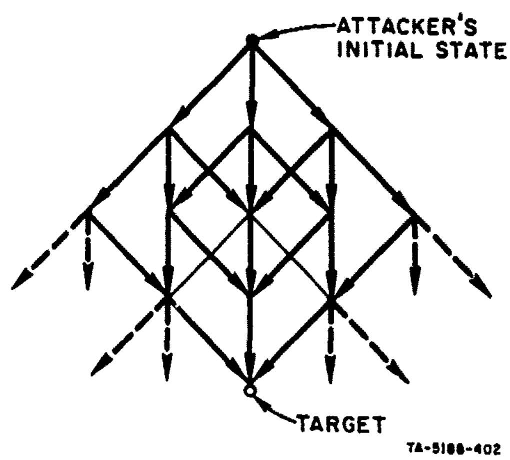

FIG. 2 SET OF ATTACKER TRAJECTORIES WHERE INITIAL STATE IS FOUR UNITS ABOVE TARGET

Example 1 (One AMM) (see Fig. 3.)

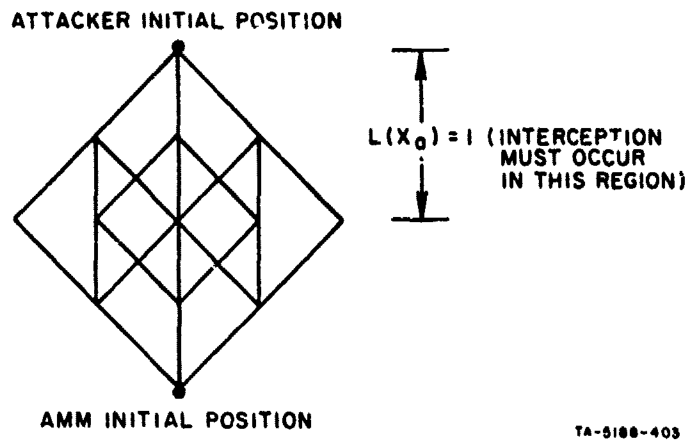

FIG. 3 EXAMPLE 1: ONE AMM WHERE INTERCEPTION MUST OCCUR AT LEAST TWO UNITS ABOVE GROUND

Solution: The optimal interception probability is 1/3 (value of the game).

(1) Interceptor optimal strategy

a. The first control is random with probability 1/3 for each move

b. The second control is to move toward the side where the attacker's first control lies.

(2) Attacker optimal strategy is not unique

a. The first control is arbitrary

b The second control is random with probability 1/3 for each move

All the solutions to the problem are convex combinations (*i.e.*, probability distributions) of three basic mixed solutions that may be graphically described as in Fig. 4. Observe that the same *control law* for the interceptor is used in the three solutions, and the different trajectories obtained are due to the response of this control law to different behaviors of the attacker. This control law guarantees the interceptor success with probability at least 1/3, no matter what the attacker does. Also any of the given randomized attacker's trajectories guarantees him passage to the defended point with probability at least 2/3, no matter what the interceptor does.

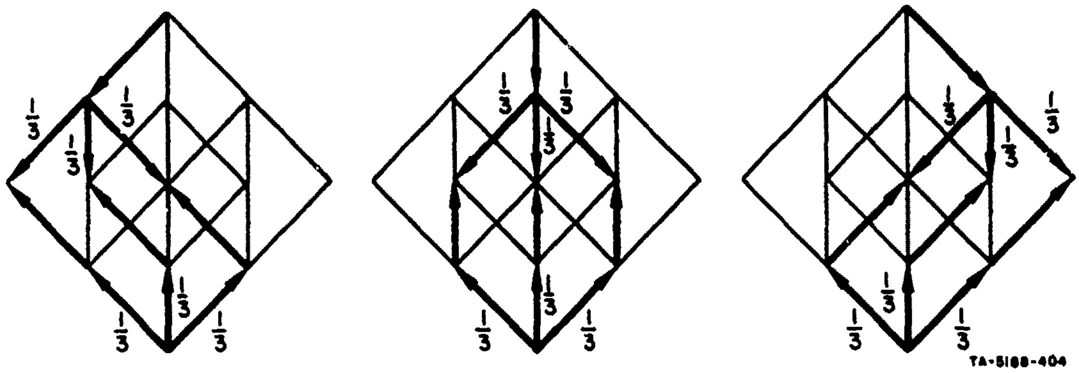

FIG. 4 THREE BASIC MIXED STRATEGIES FOR ATTACKER (Example 1)

Now, for comparison, relax the basic assumption regarding the blindness of the attacker, thus turning the problem into a discretized differential game. The optimal strategy for both sides will now be the same control law:

(1) If both missiles are on the same vertical line, take each possibility with probability 1/3.

(2) If not, then

a. Attacker: Turn away from AMM direction

b. Interceptor: Does not matter as he is lost anyway.

The value of the game is now only 1/9, *i.e.*, the interceptor cannot succeed with probability greater than 1/9.

Example 2 (One AMM)(see Fig. 5.)

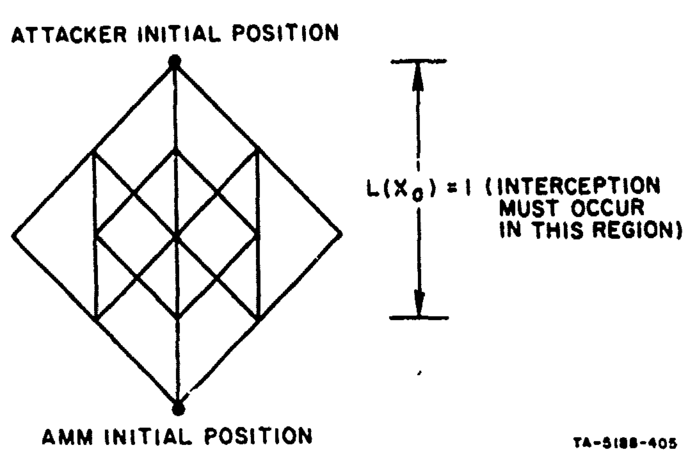

FIG. 5 EXAMPLE 2: ONE AMM WHERE INTERCEPTION MUST OCCUR AT LEAST ONE UNIT ABOVE GROUND

Note that the loss function differs from that of Example 1.

Solution: The optimal interception probability is 3/7 (value of the game).

(1) Interceptor optimal strategy

a. With probability 1/7 go straight to the center, ignoring what the attacker is doing

b. With probability 6/7, remain in the initial position for 2 units of time, after which the control depends on the attacker's position as shown in Fig. 6:

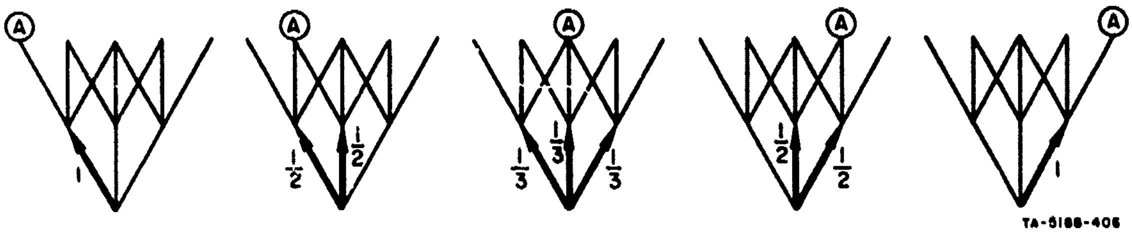

FIG. 6 INTERCEPTOR OPTIMAL CONTROL LAW WHEN INTERCEPTOR REMAINS ON GROUND FOR TWO UNITS OF TIME

(A) Indicates attacker position. The solution is unique for the interceptor.

(2) Attacker optimal strategy: the strategy is not unique; all optimal strategies ar convex combinations of the six basic solutions given by the following diagrams and their symmetric transposes. (see Fig. 7.)

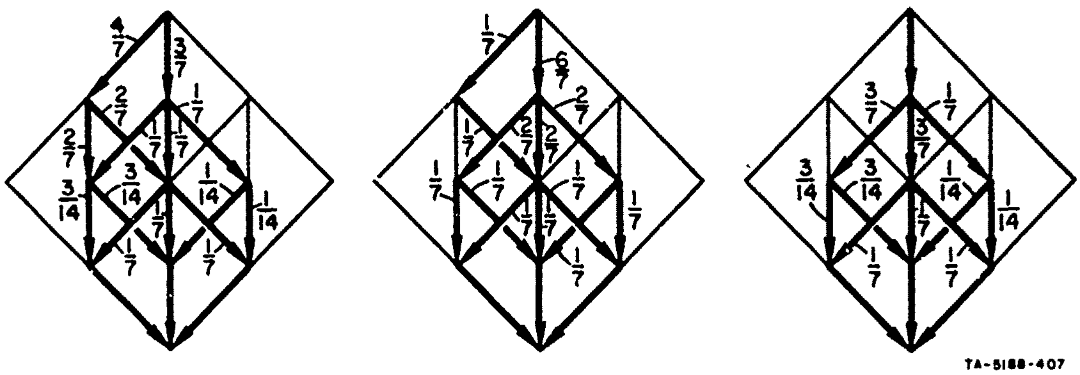

FIG. 7 SIX BASIC MIXED STRATEGIES FOR ATTACKER (Example 2)

For comparison, consider again the differential game obtained by permitting the attacker to observe the interceptor. The solution is:

(1) Interceptor's optimal control law: Wait 2 units of time, and then apply controls according to attacker's position, as before.

(2) Attacker's optimal control law:

a. If both missiles are on the same vertical line and

(*i*)-distance greater than 2, then go straight down.

(*ii*)-distance not greater than 2, then take each possibility with probability 1/3.

b. If not on same vertical line, then the control is determined from the positions in an obvious manner.

Optimal interception probability (*i.e.*, value of the game) is 1/3.

Example 3 (Two AMM's located on defended point)(see Fig. 8.)

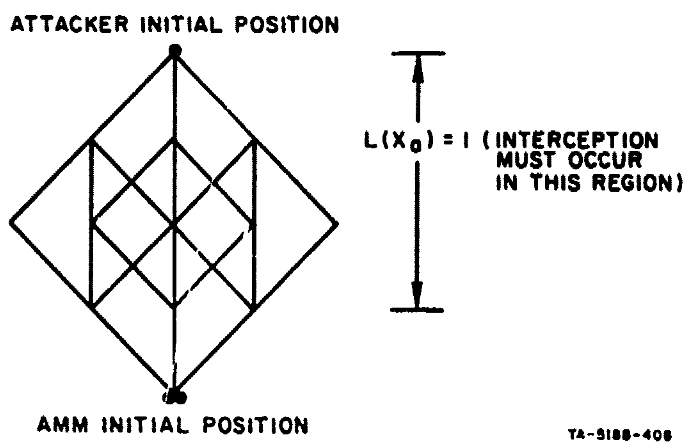

FIG. 8 EXAMPLE 3: TWO AMMs LOCATED ON TARGET WHERE INTERCEPTION MUST OCCUR AT LEAST ONE UNIT ABOVE GROUND

Solution: Optimal interception probability (value of game) is 0.8.

(1) Interceptor optimal control law: With probability 0.4 one AMM is sent straight to the center, and the other AMM waits for 2 units of time and then applies controls as in Example 2. With probability 0.6, both AMM's wait for 2 units of time, and then apply the controls represented by Fig. 9:

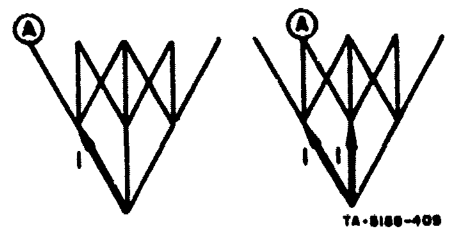

FIG. 9 INTERCEPTOR OPTIMAL CONTROL LAW WHEN BOTH REMAIN ON GROUND FOR TWO UNITS OF TIME (Example 3)

Symmetric controls for attackers on the other side. For attacker in the middle, take every pair of *different* controls with probability 1/3.

(2) Attacker optimal strategy: Not unique, and all solutions are convex combinations of the following three basic solutions (see Fig. 10.)

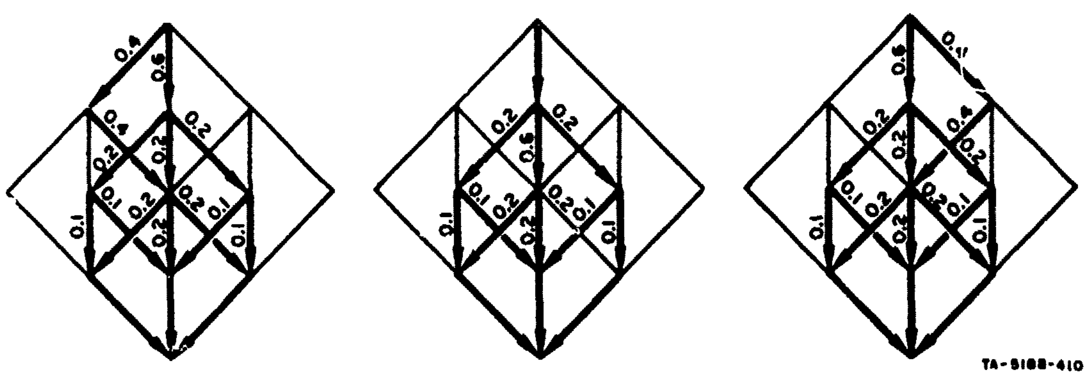

FIG. 10 THREE BASIC MIXED STRATEGIES FOR ATTACKER (Example 3)

F Conclusions

Although conclusions at this stage of development must be viewed as tentative, there are several properties of the optimal strategies in the examples treated whose consistent appearance warrants remark.

First, in all examples treated there is a substantially higher probability of interception than for the equivalent full-information differential game. This implies the possibility of a significantly more effective defense.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute 333 Ravenswood Avenue Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP N/A

3. REPORT TITLE
OPTIMAL GUIDANCE AND CONTROL IN A MISSILE DEFENSE SYSTEM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report

5. AUTHOR(S) (First name, middle initial, last name)
R. E. Larson R. M. Dressler W. G. Keckler D. G. Luenberger
L. Meier R. S. Ratner A. Shefi

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1967	104	30

8a. CONTRACT OR GRANT NO. CONTRACT DA-01-021-AMC-90006(Y)	9a. ORIGINATOR'S REPORT NUMBER(S) SRI Project 5188-233
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY NIKE-X PROJECT OFFICE U.S. ARMY MATERIEL COMMAND REDSTONE ARSENAL, ALABAMA

13. ABSTRACT

This report summarizes the work done by the Information and Control Laboratory during 1967 in the area of optimal guidance and control in a missile defense system. General problem formulations for the optimal interception of both ballistic and maneuvering reentry vehicles are given. A computer program based on dynamic programming for pre-launch calculations is described. A second computer program that utilizes the gradient method for in-flight guidance is also discussed. Finally, a game-theoretic approach to the problem of intercepting maneuvering reentry vehicles is presented.

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Optimal Control Stochastic Control Combined Optimum Control and Estimation Differential Games Game Theory Dynamic Programming Gradient Methods Kalman Filter Optimum Missile Interception Ballistic Reentry Vehicle Interception Maneuvering Reentry Vehicle Interception Optimum Guidance Prelaunch Calculation Hardpoint Defense Multiple Interceptors for a Single Target						

DD FORM 1 NOV 65 1473 (BACK)

(PAGE 2)

UNCLASSIFIED

Security Classification